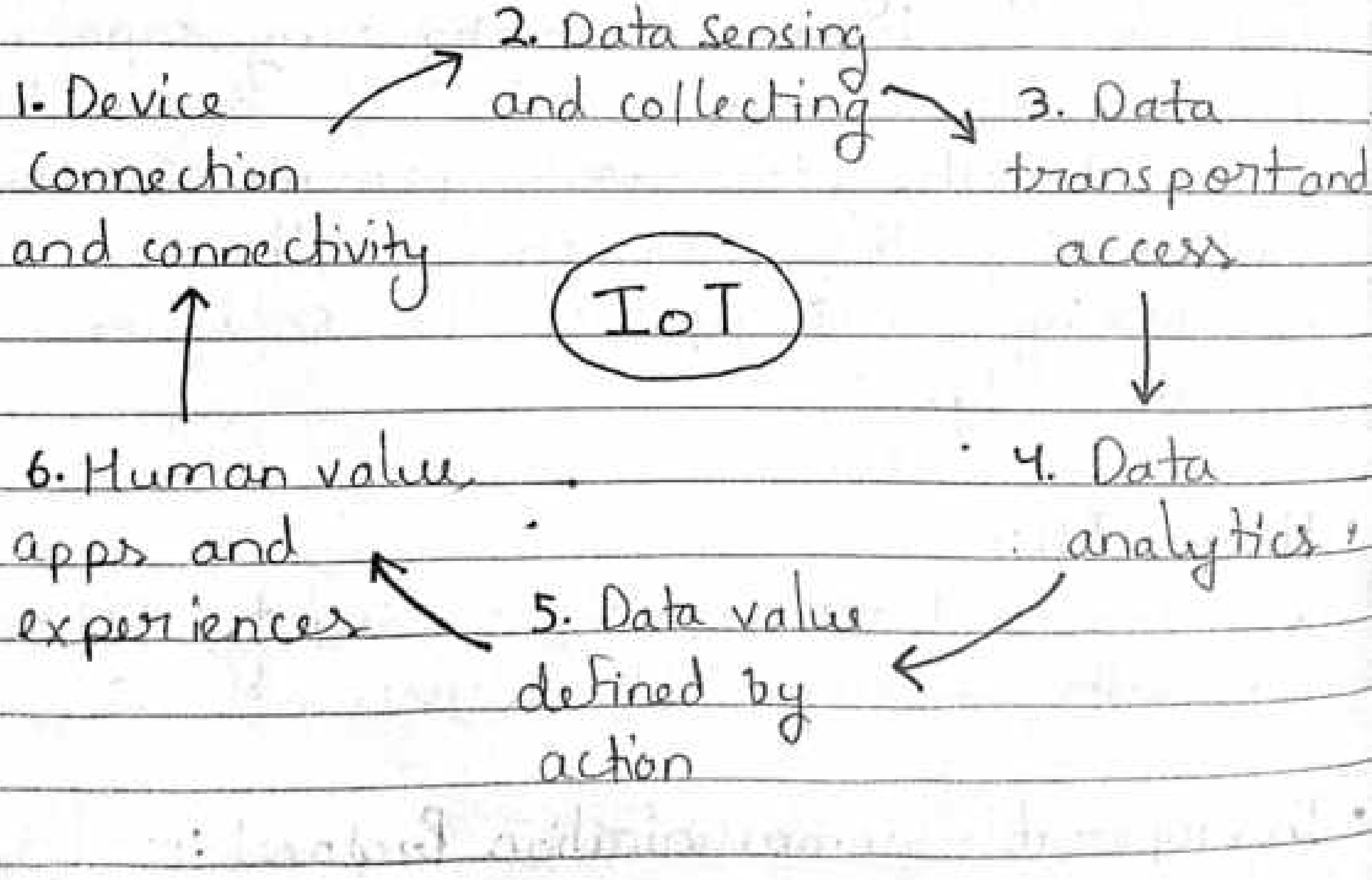
IoT (Internet of Things)

Definition :- The IoT is the network of physical objects with unique identifiers that are connected with each other and are embedded with electronics, software and sensors which enables these objects to collect and exchange data.

How all this actually happens


1. Device Connection and connectivity
2. Data sensing and collecting
3. Data transport and access
4. Data analytics
5. Data value defined by action
6. Human value apps and experiences
IoT

Characteristics of IoT

- **Dynamic and Self-Adapting :-**
IoT devices and system may have the capability to change dynamically depending upon the system and operating conditions or sensed environment.
Example- the surveillance cameras can change their modes based on day or night.

- **Self-configuring :-**
IoT devices have self-configuring capability which allows large number of devices to work together, ~~to work~~ provide certain functionality they can change their networking and update the software automatically.

- **Unique ID :-**
IoT devices have a unique identity differentiated with ~~tt~~ unique IP address.

- **Interoperable Communication Protocol :-**
IoT devices can communicate with number of interoperable (communicate with other devices without special effort) communication protocols

- Integrated into information Network:-

IoT devices are integrated into the information network that allows them to communicate and exchange data with other devices and systems.

- Connectivity:-

Things in I.OT should be connected to the ~~infrac~~ infrastructure.

- Intelligence:-

Extraction of Knowledge from the generated data is important, ~~sensor generated data and this data should be~~

- Scalability:-

IoT devices should be designed in such a way that they can be scaled up or down easily on demand.

Design of IoT

→ Physical design of IoT

→ Logical design of IoT

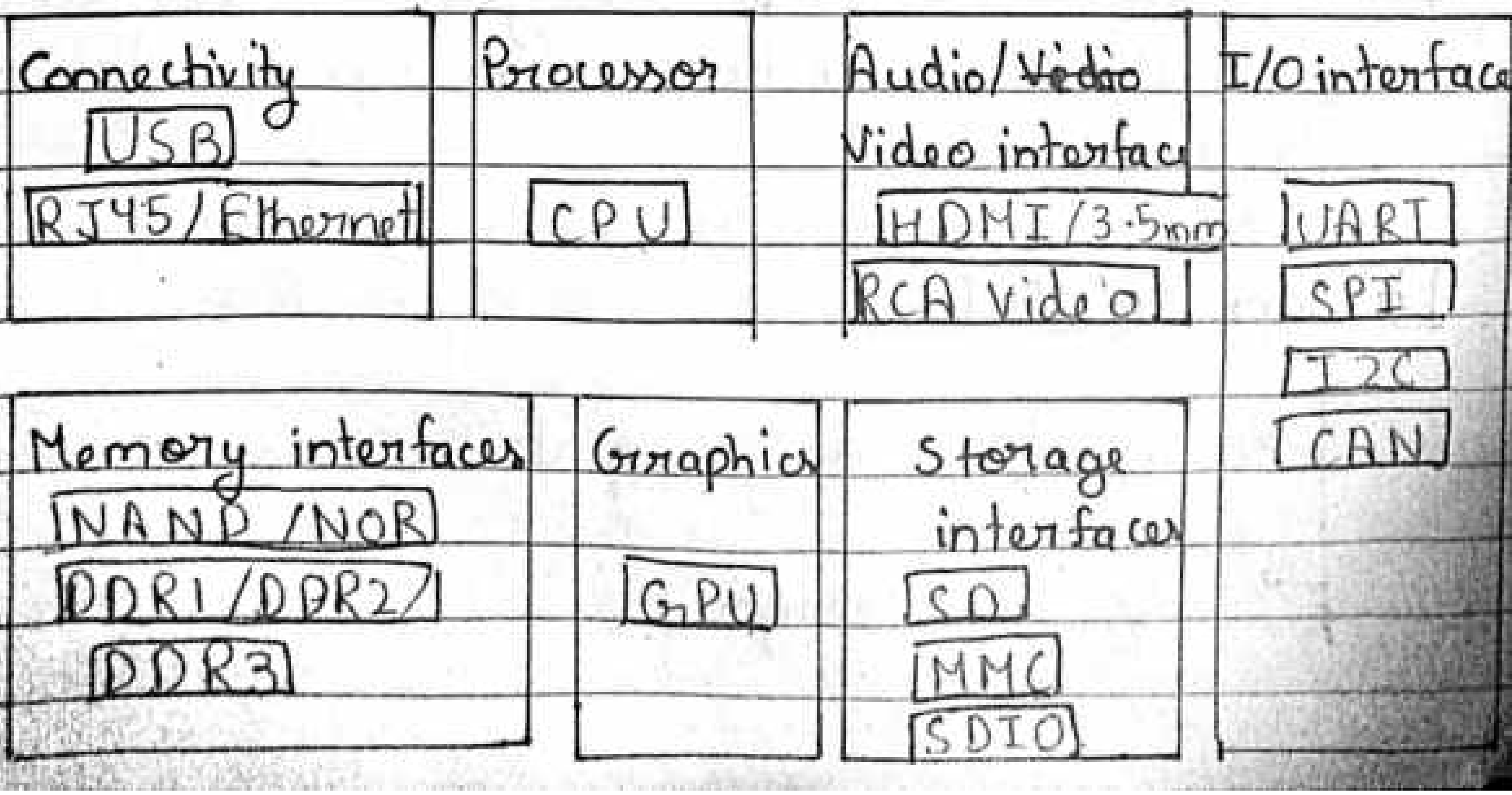
1) Physical Design of IoT :

The physical design of IOT consists of the
→ things in IoT
→ IoT protocols & layers

- Things in IoT :

The word 'things' refers to the IoT devices which have unique ~~identifiers~~ identities and can perform remote sensing, actuating and monitoring capabilities. :
These devices can exchange data and communicate with each other.

The IoT devices consists of several interfaces.



· IoT Protocols

Application
Transport
Network/Internet
Link

1) Link layer

This protocol determines how the data is physically sent over the network layer ~~ex~~. It determines how the packet are coded and signaled by the hardware device over the medium to which the host is attached.

Example:- IEEE 802.3 - Ethernet (Wired-connection)
802.11 - WiFi
802.16 - WiMax
2G/3G/4G - Mobile communication

2) Network/ Internet Layer

The network layers are responsible for sending of IP datagram's from the source network to the destination network. It performs host addressing and netw packe routing. The datagrom's consists of source and destination addresses where host identifies using IP schemes as IPV4 and IPV6.

IPV4: It is used to identify the devices on a network using ~~hierarichal~~ hierarchical addressing scheme. It uses 32 bit addresses that allows total 2^{32} or 4 billion devices.

IPV6: It is the new version of internet protocal which uses 128 bits of addresses that allows 2^{128} or 3×10^{28} addresses.

3) Transport layer

The transport layer protocols provide end to end message transfer capability. ~~Independent~~ of the underlying network. The message transfer capability can be set up on connections, either using handshakes (TCP) or without handshakes / acknowledgements (UDP). The transport layer provide functions such as error control, segmentation, flow control and congestion control.

TCP - (Transmission Control Protocol)

→ Connection - oriented pricotocol

→ provides extensive error checking.

→ reliable as it guarantees delivery of data to destination router.

- → Sequencing of data
- → slower than UDP.
- → Retransmission of lost packets.

header-20-80 bytes

UDP (User Datagram Protocol)

- → connectionless
- → basic error checking
- → no retransmission
- → does not gurantee delivery
- → no sequencing of data
- → header - 8 bytes

4) Application Layer

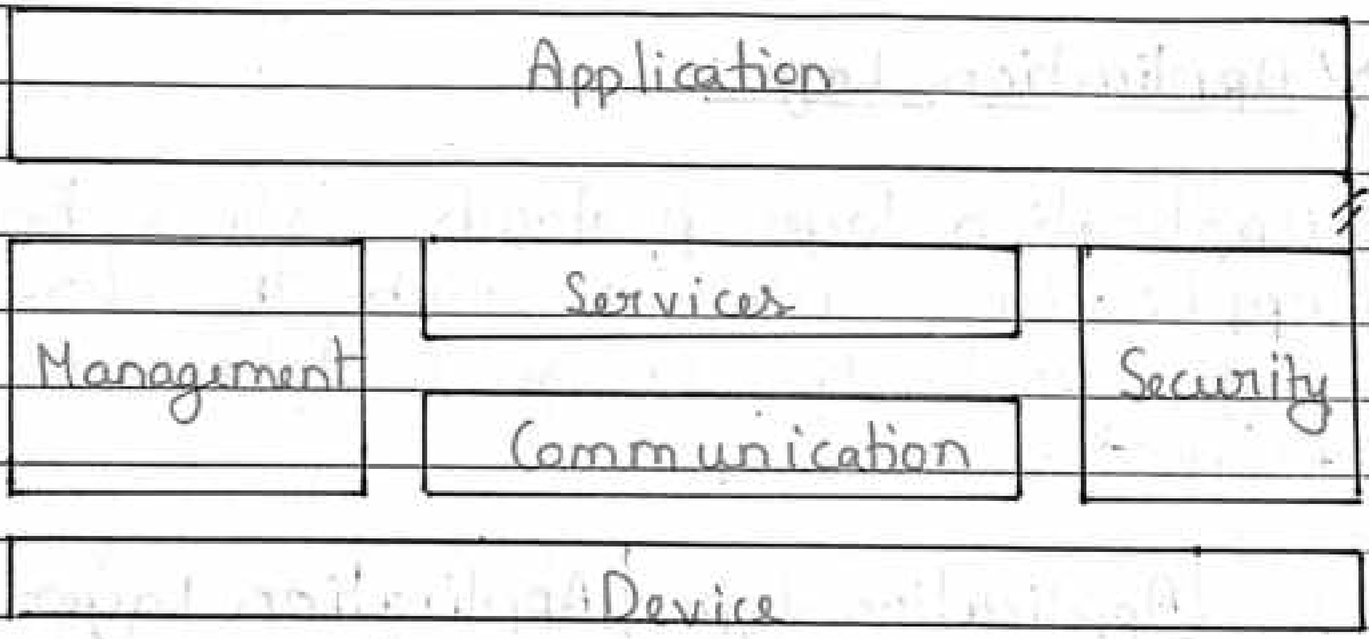
Application layer protocols defines how the applications interface with the lower layer protocols to send data over the network.

Application Type	Application Layer Protocol
Electronic mail	Send: SMTP (Simple Mail Transfer Recieve: POP3 (Post Office P)
M2M	CoAP (Constrained App. P)
WWW	HTTP (Hyper Text Transfer P)
File transfer	FTP, TFTP (Trivial FTP)
Internet telephony	Proprietary

2) Logical Design of IoT

Logical design of an IoT describes about abstract representation of the entities and process without going to low level specifies of the implementation.

→ IoT functional blocks



- Device - IoT devices which provide sensing, monitoring and control functions.

- Communication - handles communication for IoT system

- Services- deals with services such as device monitoring. d. control services and d. discovery.

- Management - used to monitor the complete IoT system.

- Security - provide security by providing the functions such as authentication, authorization and data security.

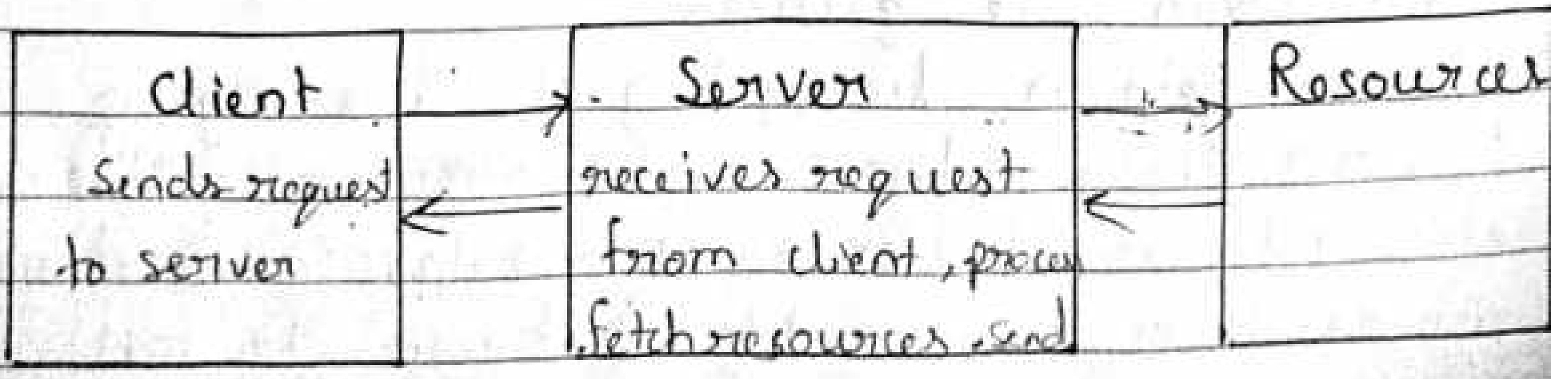
- Application - IoT applications provide an interfac that the users can use to control and monitor various aspects of an IoT system.

→ IoT Communication Model

- Request - Response

→ In Request-response communication model client sends request to the server and the server responds to the request.

→ When the server receives the request it decides how to respond, fetches the data, retrieves resources, and prepares the response and sends to the client.



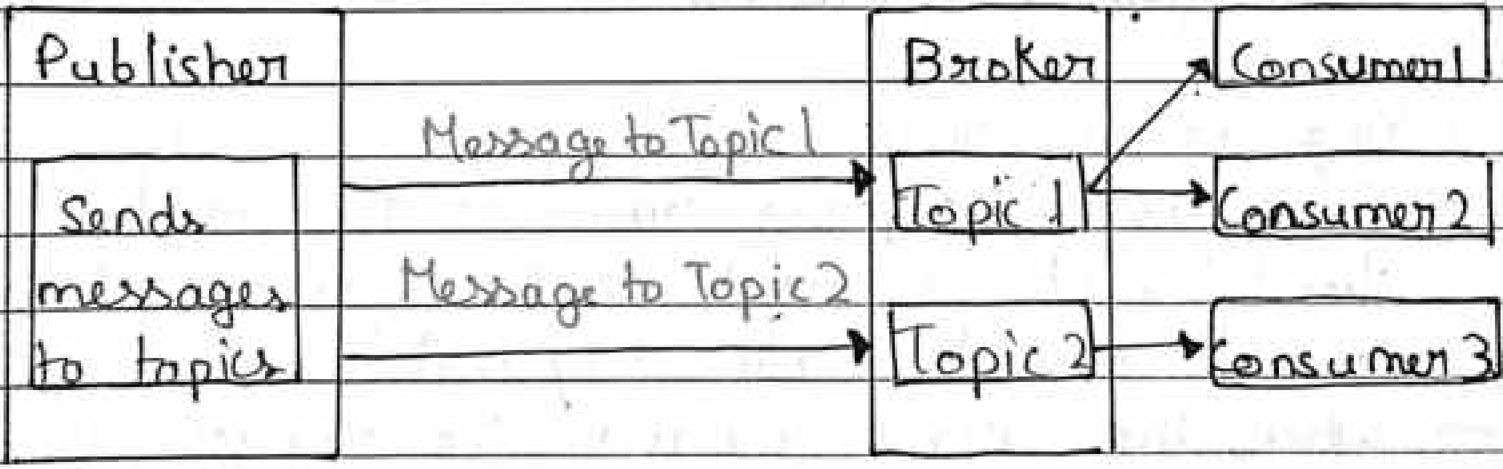
- Publish - Subscribe

→ This model involves publishers, brokers and consumers.

→ Publishers are the sources of data. So It sends the data to the topic which are managed by the broker. They are not aware of consumers.

→ Consumers subscribe to the topics which are managed by the broker.

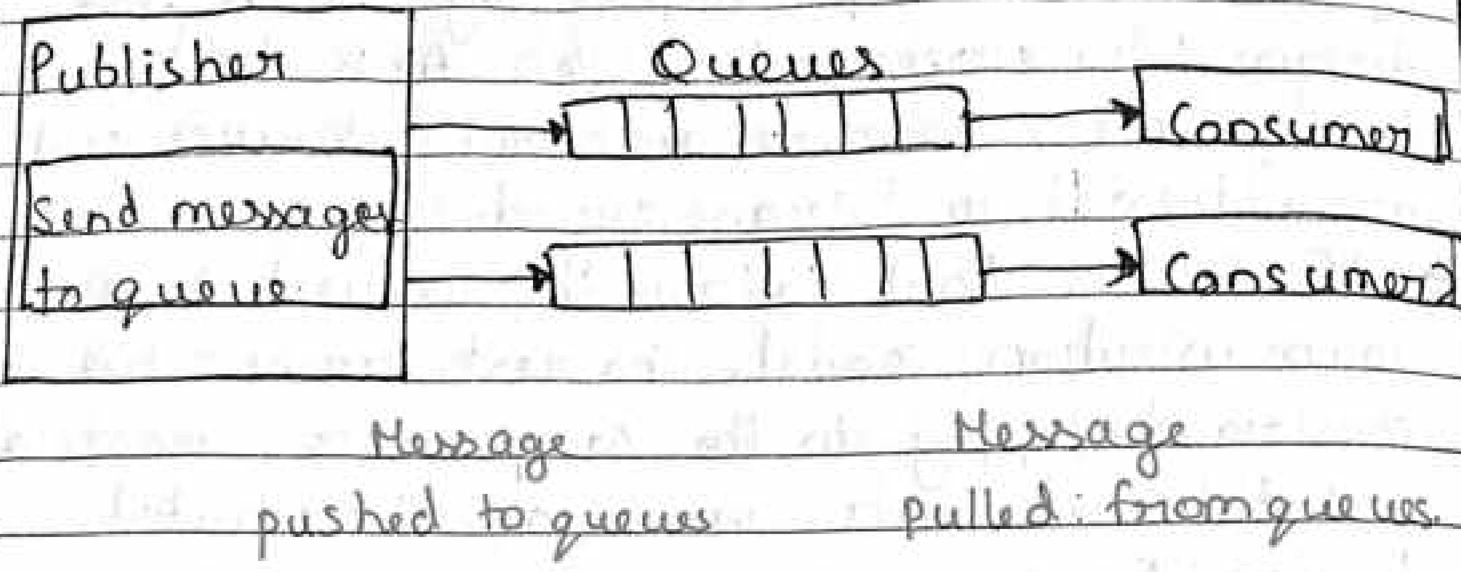
→ When broker receives the data from the publisher, it sends to all the consumers.



- Push-Pull

→ In this model the publisher push the data in queues and the consumers pull the data from the queues.

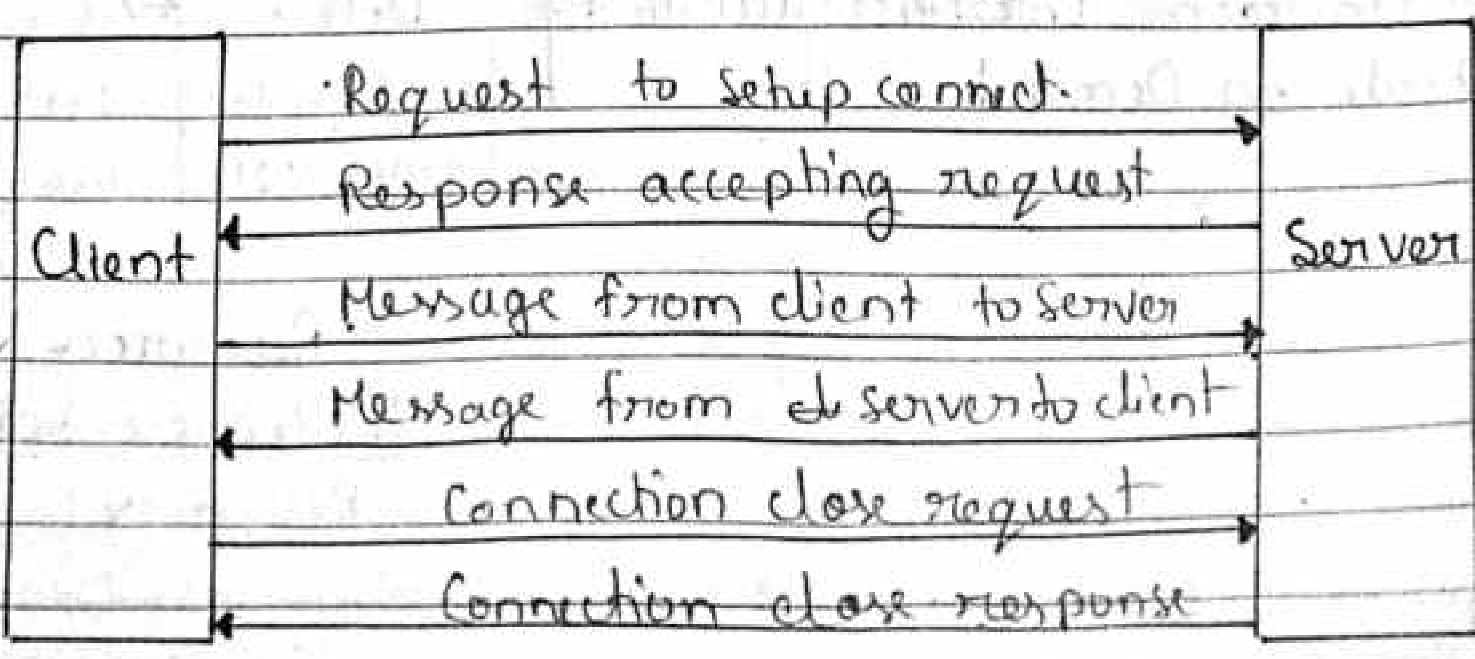
→ Queues help in decoupling the messaging between the producer and consumers. Queues also act as buffer which helps in situation where there is mismatch between the rate at which the producers push the data and consumers pull the data.



- Exclusive Pair

→ It is a bi-directional, fully duplex communication model that uses a persistent connection between the client and server, once the connection is established it remains open until the client sends a request

→ Both can send message to each other.


Client
Server
Request to setup connect.
Response accepting request
Message from client to server
Message from ~~cl~~ server to client
Connection close request
Connection close response

IoT Communication API

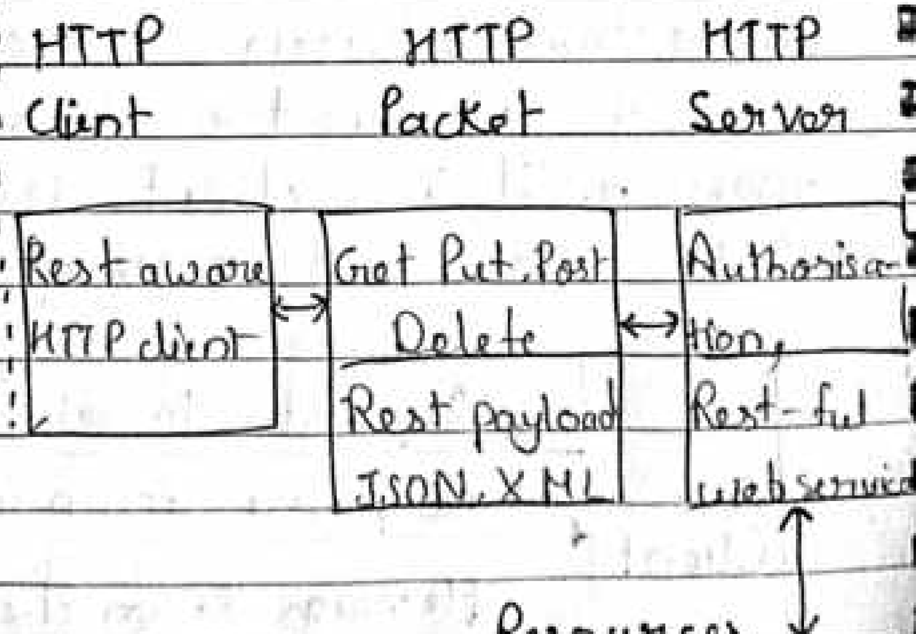
• REST Based G.A.C.A.

→ Representational state transfer (REST) is a set of architectural principles by which you can design Web services the Web APIs that focus on system's resources and how resource states are addressed and transferred.

→ REST APIs that follow the request response communication model, the rest architectural constraint apply to the components, connector and data elements, within a distributed hypermedia system.

→ The rest architectural constraint are as follows:

1) Client-Server
2) Stateless
3) Cache-able
4) Layered System
5) Uniform Interface
6) Code on Demand.



- Websocket based Communication API

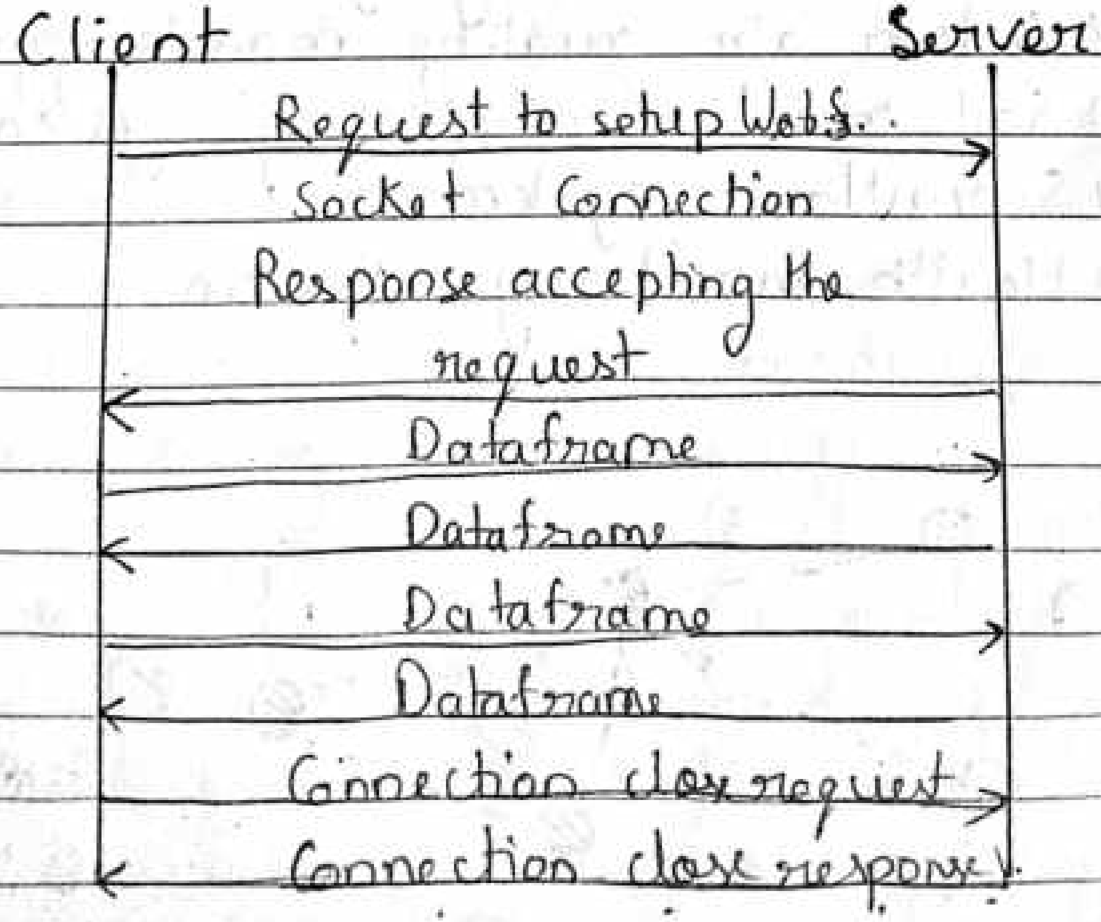
→ It allow bi-directional, full duplex communication between clients and servers.

→ Websocket APIs follow the exclusive pair communication model.

→ W.C. begins ~~#~~ with a ~~com~~ connection setup request sent. by the client ~~or~~ to server.

→ ~~7~~ If the server supports websocket protocol the server responds to the websocket handshake response.

→ ~~After the~~ Websocket API reduce the network traffic and ~~latency~~ letency as there is no overhead for connection setup and termination requests requirements.



IoT enabling Technologies

- Wireless Sensor Network (WSN)

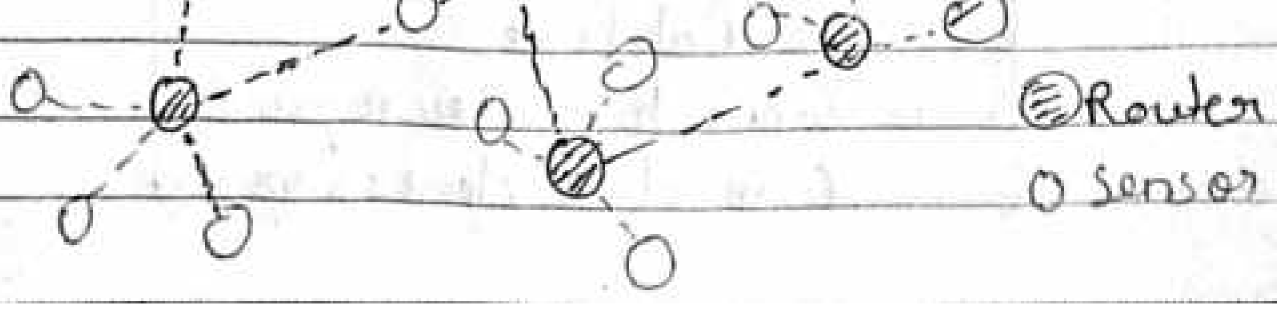
→ A WSN comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions.

→ A WSN consists of end nodes, routers and coordinator.

→ End nodes have several sensors attached to them where the data is passed to coordinator. with the help of routers.

→ The coordinator also acts as gateway that connects WSN to Internet.

Example- 1) Weather Monitoring System
2) Indoor air quality monitoring system
3) Soil moisture monitoring system.
4) Survillance systems
5) Health monitoring system.



- Cloud computing

→ It is the delivery of different services through the internet, including data storage, servers, databases, networking and software.

→ Characteristics:-

- Broad network access
- On demand self service (can provision additional resources on your own).
- Rapid Scalability
- Measured service (Pay on only services used).

→ Provides different services such as:-

- IaaS (servers, networking, storage, and data center space on a pay per use basis.

- PaaS (provides a cloud based environment with everything required to support the complete life cycle of building and delivering web based (cloud) applications - without the cost and complexity of buying and managing underlying hardware, Software, provisioning and hosting)

- SaaS (is a way of delivering app. over the internet as a service. Instead of installing and maintaing software, you simply access it via the Internet, freeing yourself from complex software and hardware managent.

Saas applications are sometimes called Web based software, on demand software or hosted software.

Saas applications run on a SaaS provider's servers and they manages security, availability and performance.

- Big Data Analytics

→ It refers to the strategy of analyzing large volumes of data or big data.

→ Big data is gathered from a variety of sources including social networks, videos, digital images, sensors and sales transaction records.

→ Several Steps involves in analyzing big data are - data cleansing, munging, processing & and visualization.

Example- 1) Sensors data generated by W.M.S
2) Data generated by IoT systems for locations and tracking of vehicles.
3) Sensors embedded in industry and energy system.

4) Health and fitness data generated by IoT system such as fitness bands.

- Embedded Systems.

→ It is a combination of hardware and software system used to perform special tasks.

→ It includes microcontroller/microprocessor memory (RAM, ROM), networking units (Ethernet, wifi adapters), input/output units (display, Keyboard etc) storage system (Flash memory)

→ It collects the ~~ad~~ data & sends it to internet.

Iot Levels and Deployment Templates

Iot system consists of following components.

Device- An IoT device allows identification, remote sensing, remote monitoring capabilities.

Resource- Software components on IoT device for-

→ accessing, processing and storing data.

→ ~~onthe~~ Controlling actuators.

→ Enabling network access for the device

Controller Service - Sends data from the device to the web service and receives commands from the application for controlling device.

Database - can be local or cloud and stores the data generated by the IoT device.

Web Service - Serve as a link between the IoT device, application, database and analysis components.
→ Can be implemented by HTTP &REST princip or using Web Socket protocol.

Analysis Component - responsible for analyzing the IoT data and generating results in the form that is easy to understand for user.

Application - provides an interface that the user can use to control and monitor various aspects of the IoT system.

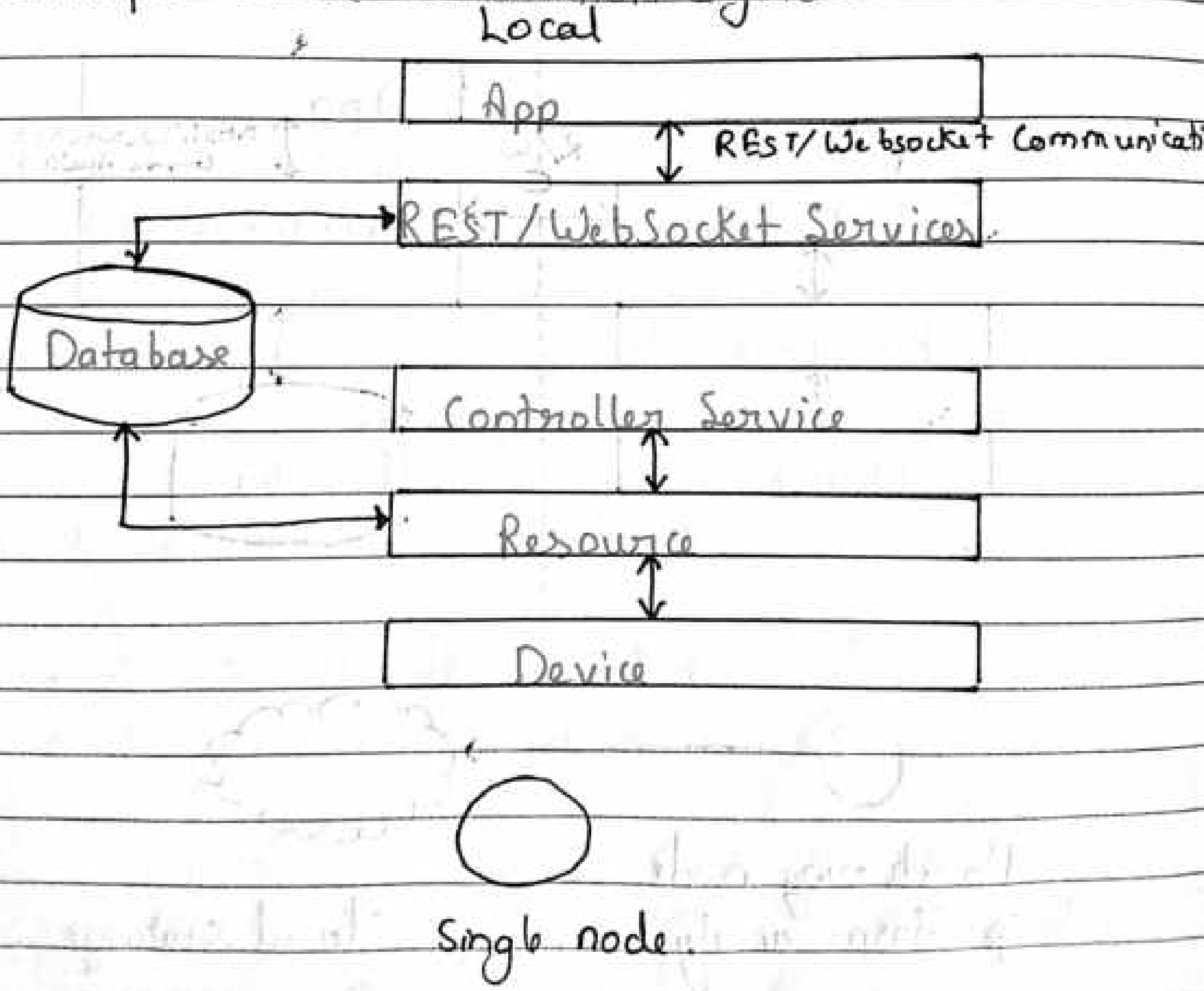
→ It also allows users to view ~~theo~~ the system status and the processed data.

IoT level-1

→ It has a single node/ device that perform sensing, stores data, perform analysis and hosts the application.

→ Suitable for modelling ~~low level~~ low cost and low complexity solutions. where the data involved is not big & he analysis requirements are not computationally intensive.

Example- Home automation system.



IoT Level-2

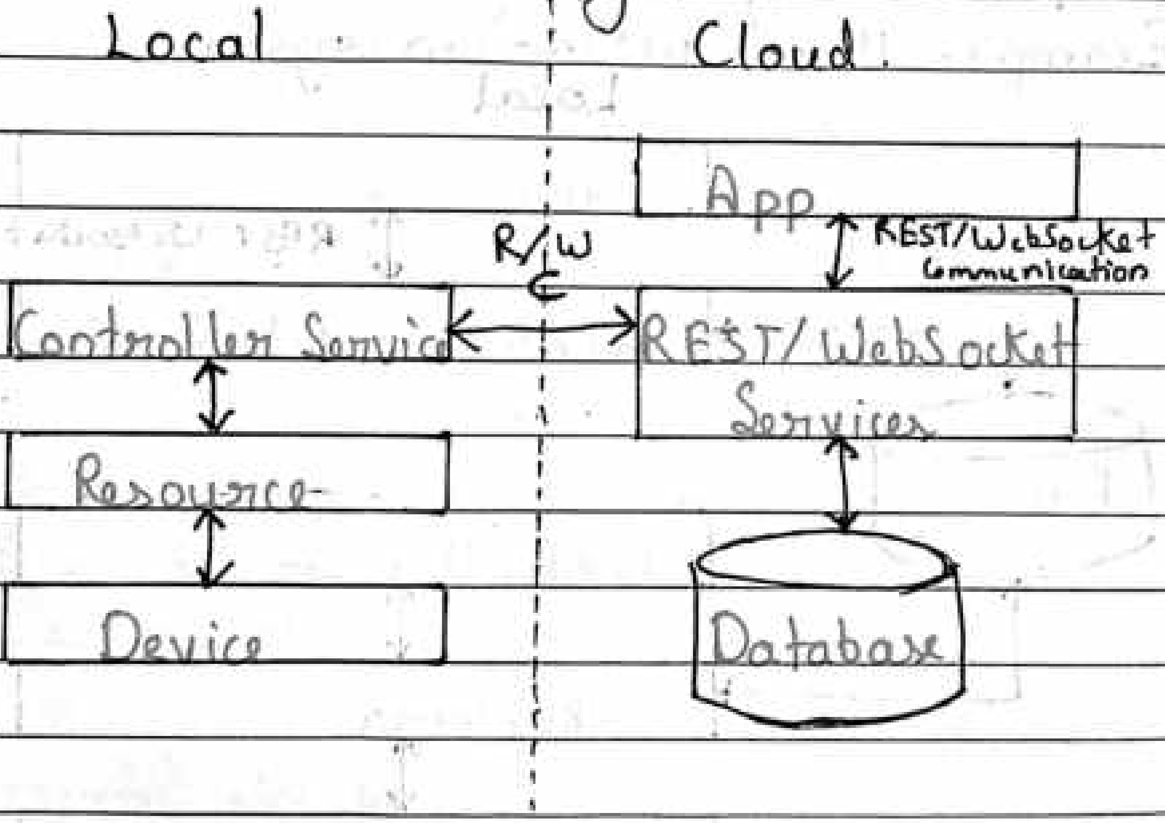
→ Single node

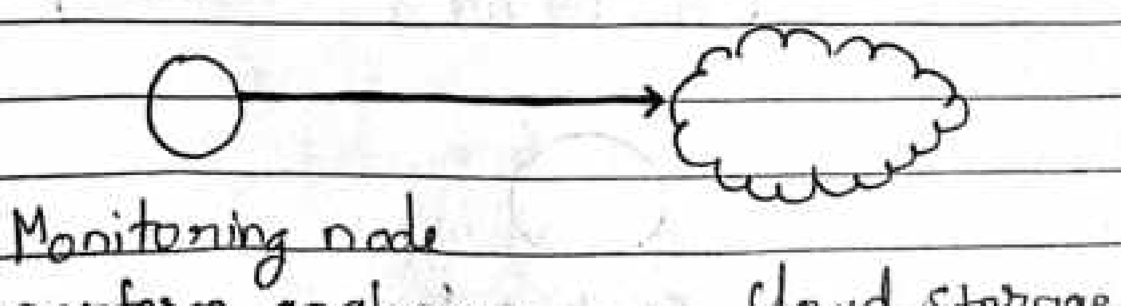
→ Data is stored in the cloud & the app is usually cloud based.

→ Data involved is big.

→ Not computationally intensive.

Example- ~~Sa~~ Smart Irrigation

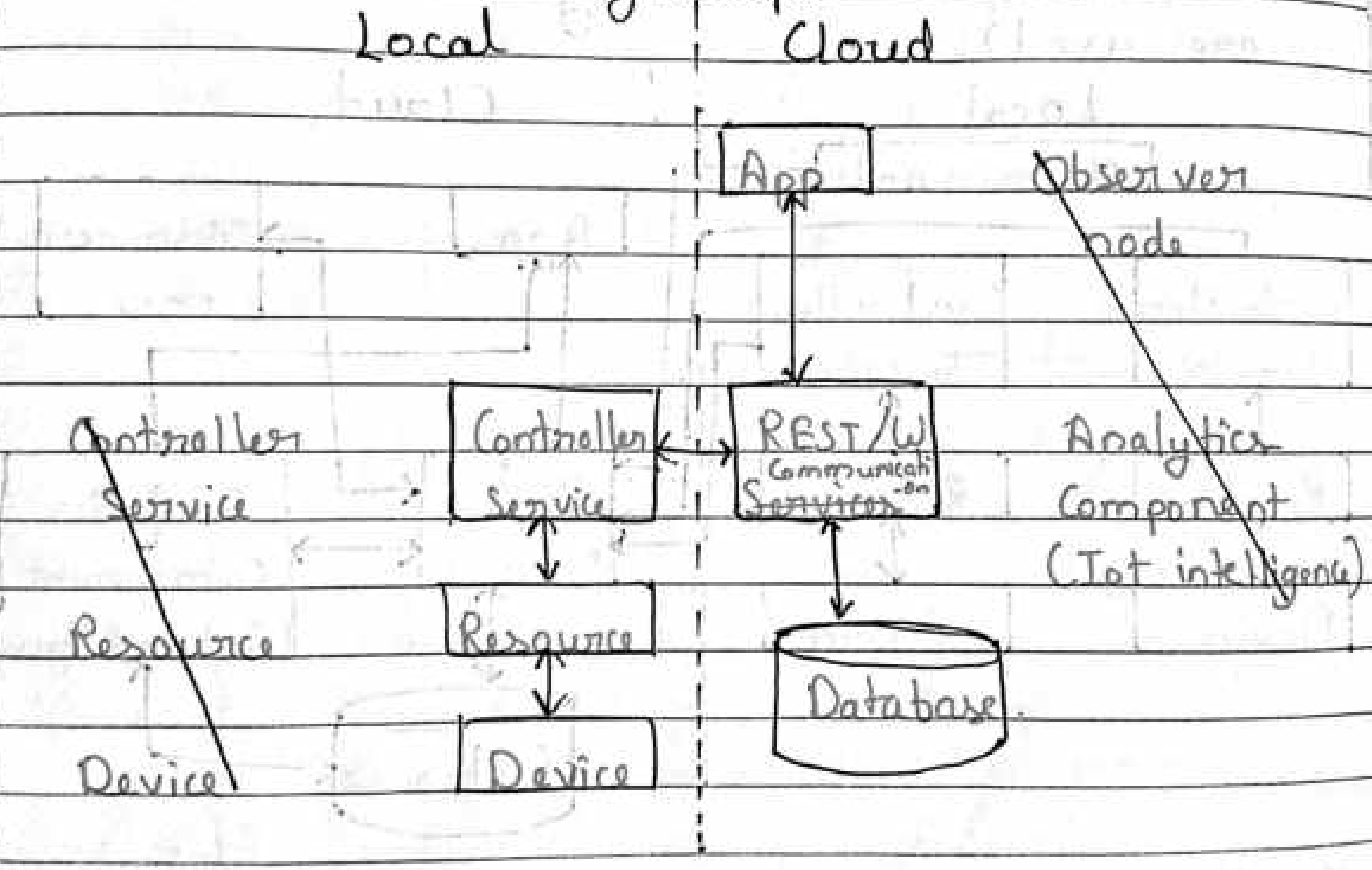


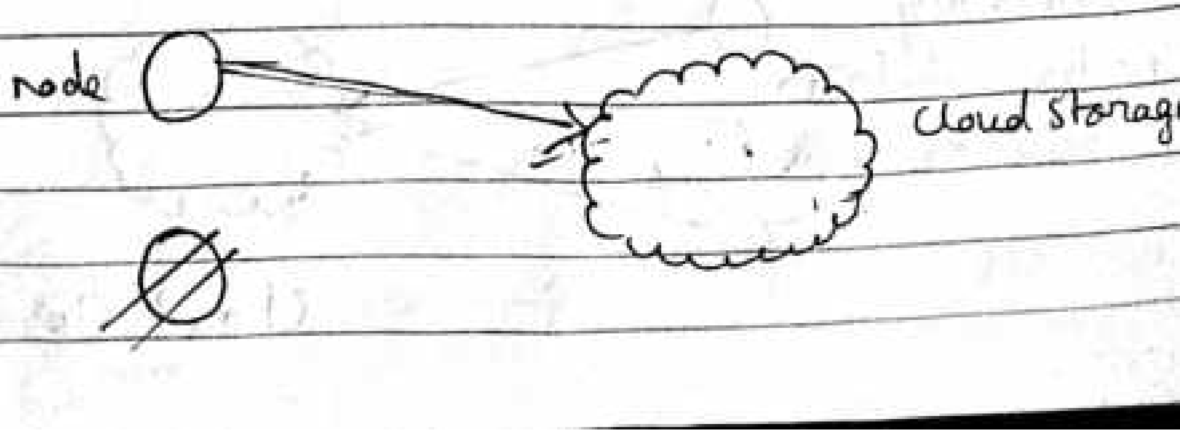


IoT Level 3

-> Single node, stored in cloud
-> Data involved is big, an
-> analysis requirements are computationally intensive.

Example- Tracking package handling
Sensors used- Accelorometer
Gyroscope





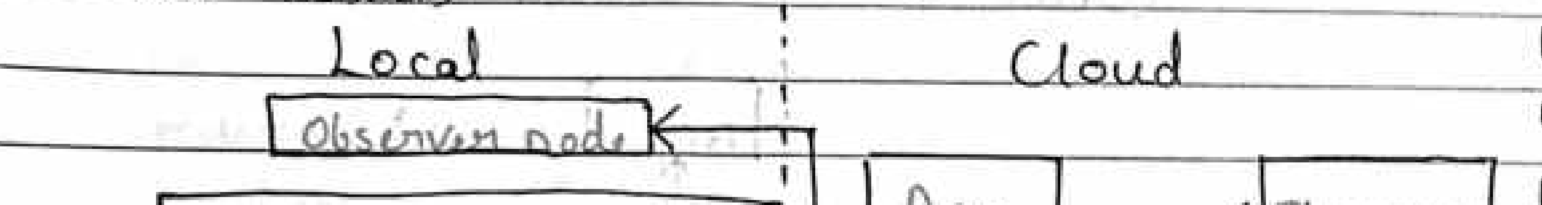
IoT Level-4

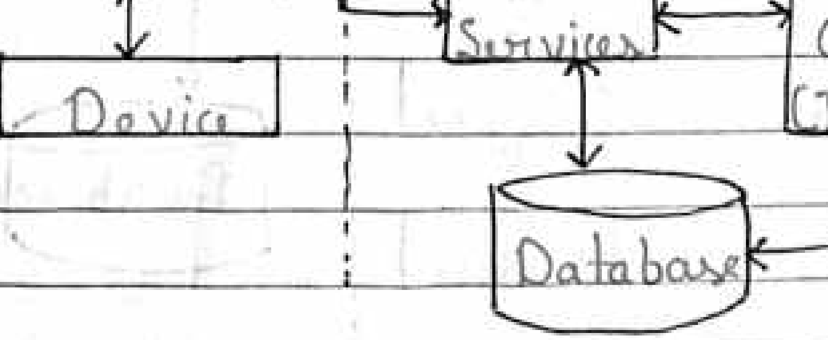
→ Multiple nodes perform local analysis

→ Contains local & cloud based observer nodes which can subscribe and receive information collected in the cloud from IoT devices

→ data big, C. I.

Example- Noise monitoring (Sound sensors are used)





IoT Level-5

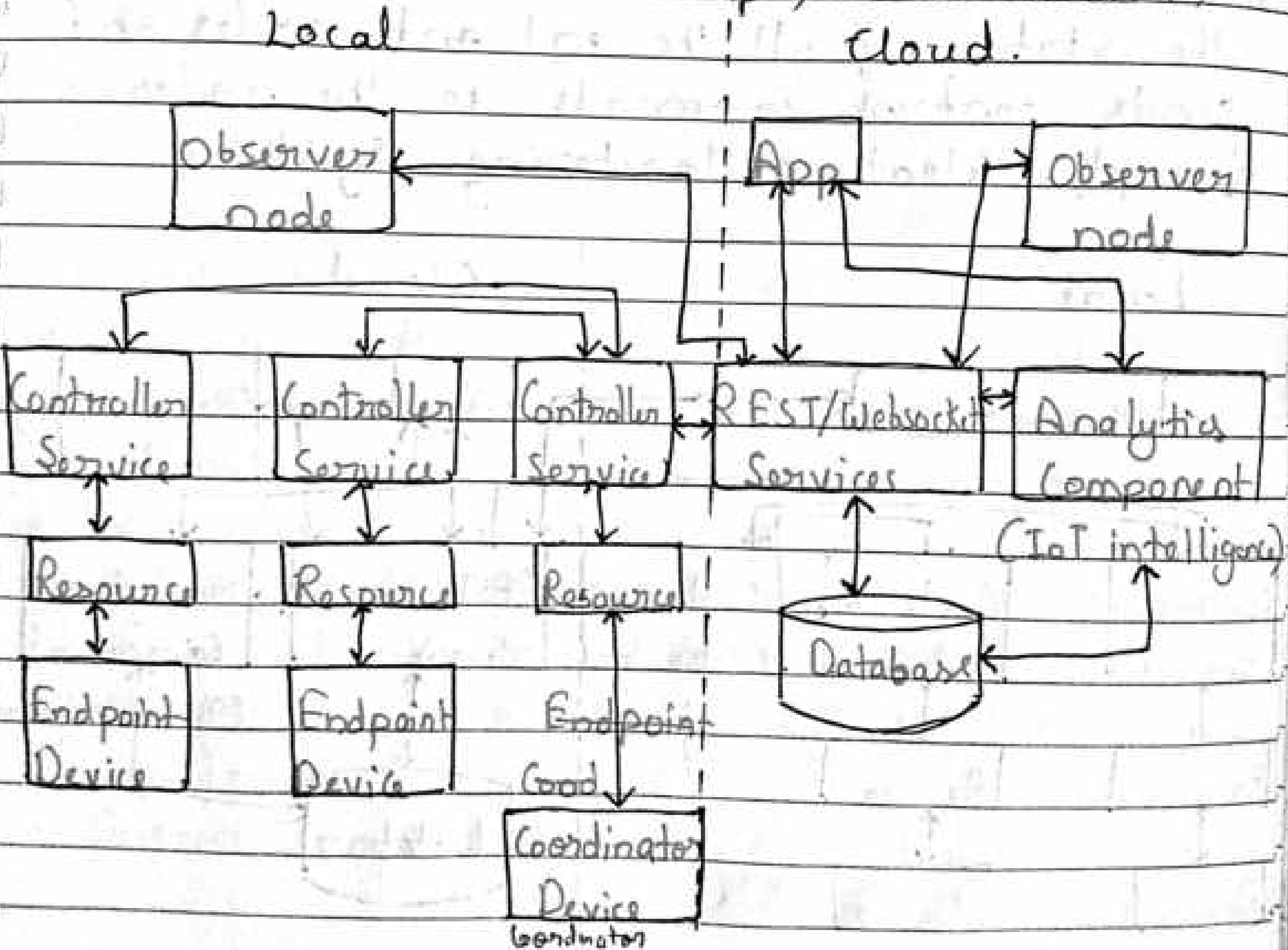
→ Multiple nodes and one coordinator node.

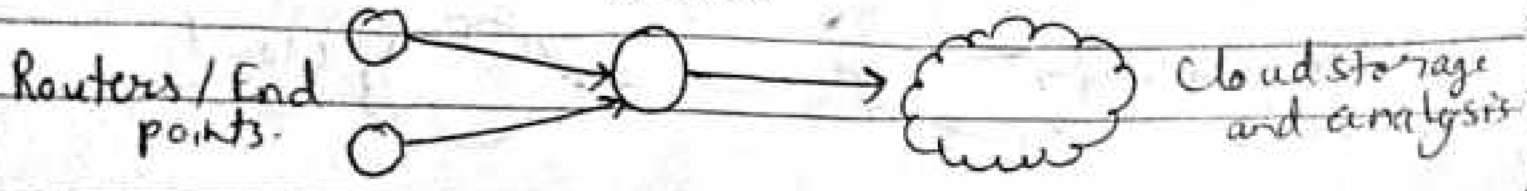
→ Coordinator node collects data from the end nodes and sends it to the cloud.

→ ~~Data~~ Level5 IoT systems are suitable for solutions based on wireless sensor networks.

Example - Forest fire Detection

Sensors measure the temp., smoke, weather.





IoT Level-6

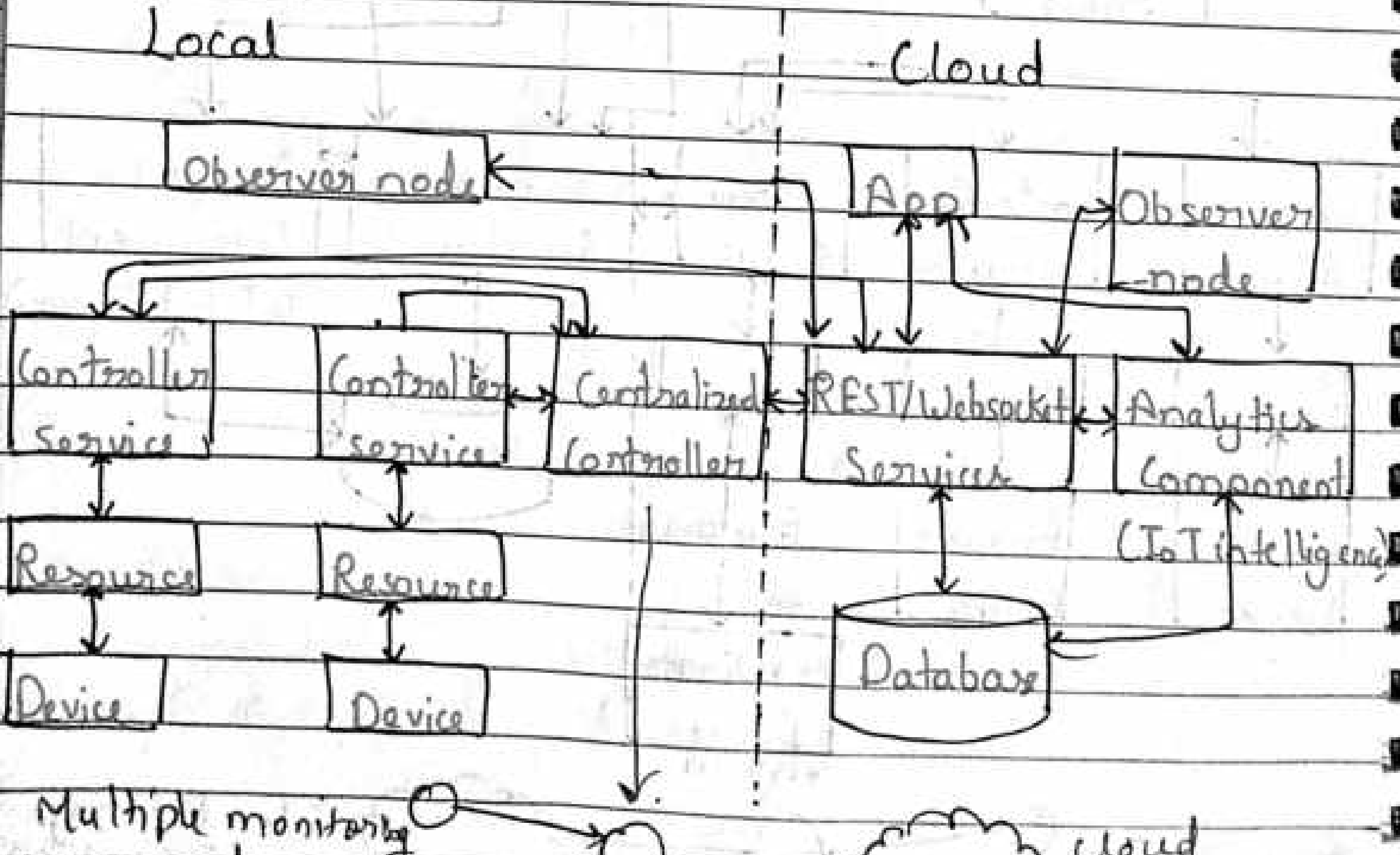
→ Multiple nodes that perform sensing and/or actuation and send data to the cloud.

→ The analytics component analyzes the data and stores the results in the cloud database.

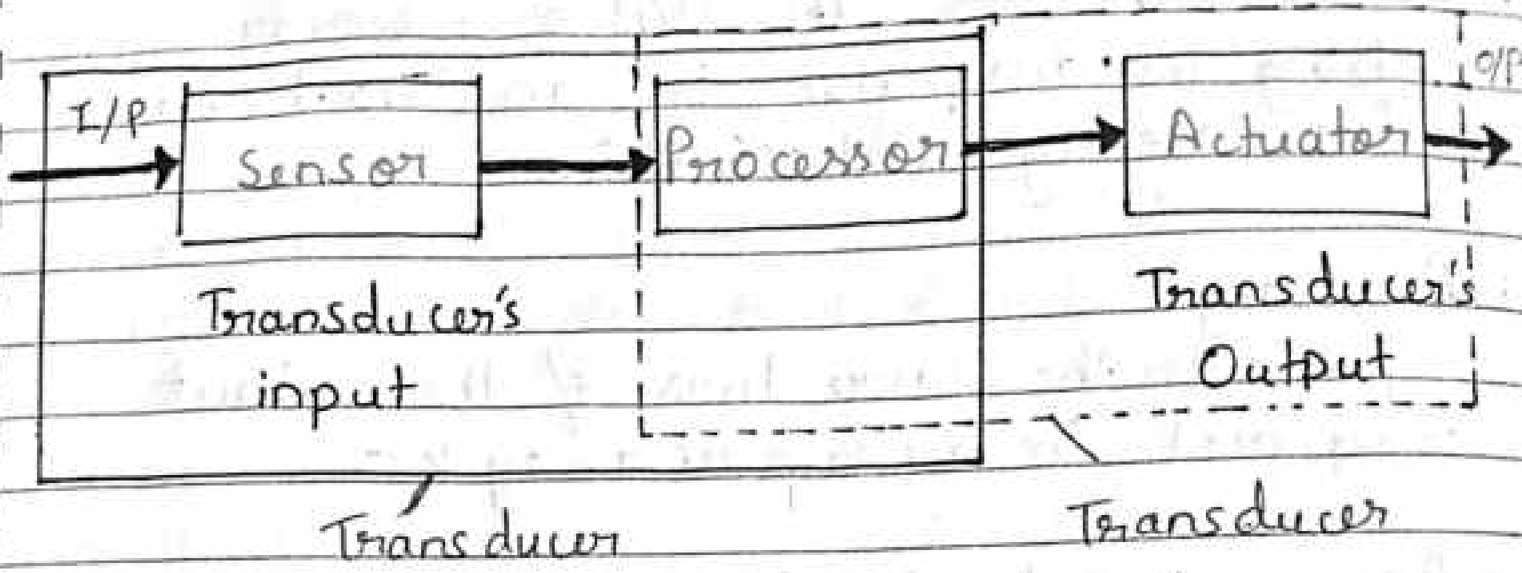
→ The results are visualized with the cloud based application.

→ The centralized controller is aware of the status of all the end ~~and~~ nodes and sends control commands to the nodes.

Example- Weather Monitoring System.



IoT Hardware



Transducer: Converts a signal from one physical form to another form.

Sensor

- The characteristic of any device or material to detect the presence of a particular physical quantity.
- The output of sensor is signal, which is converted to human readable Form like changes in V. characteristics, changes in resistance, capacitance, C. characteristics, impedance etc.
- Sensor characteristics
  - → Static
  - → Dynamic

Static Characteristics

- It is about how the O/P of a sensor change in response to an input change after steady state condition.

1) Accuracy

→ Represents the correctness of the output compared to a superior system

→ Acc. = Standard value − Measured value.

2) Range

→ Gives the highest and the lowest value of the physical quantity ~~wh~~ within which the sensor can actually sense.

→ Beyond this value there is no sensing or no kind of response.

3) Resolution

→ Provides the smallest change in the input that a sensor is capable of sensing.

→ Resolution is an important specification towards selection of sensors.

→ Higher the resolution better the precision.

4) Errors

The difference between the standard value & the value produced by sensor.

5) Sensitivity

→ Sensitivity indicates ratio of incremental change in the response of the system with respect to incremental change in input parameter.

→ It can be found from slope of output characteristics curve of a sensor.

6) Linearity

→ The deviation of sensor value curve from a particular straight line.

7) Drift

The difference in the measurements of sensor from a specific reading when kept at the value for a long period of time.

8) Repeatability

The deviation between measurements in a sequence under same conditions.

Dynamic Characteristics

Properties of the system's transient response to an input.

1) Zero order System

→ Output shows a response to the input signal with no delay.

→ Does not include energy-storing elements

→ Ex. Potentiometer measures linear and rotary displacements.

2) First order system

→ When the output approaches its final value gradually.

→ Consists of an energy storage and dissipation element.

3) Second order system

→ Complex output response

→ The output response of sensor oscillates before steady state.

Sensor Classification

→ Passive & Active

→ Analog & digital

→ scalar and vector.

- Passive Sensor.

Cannot independently sense the input.
Example- Accelerometer, soil moisture, water level, and temperature sensors.

- Active Sensor

Independently sense the input
Example- Radar, sounder and laser altimeter sensors.

- Analog Sensor

The response or output of the sensor is some continuous function of its input parameter
Example: Temperature sensor, LDR, analog pressure sensor, and analog hall effect.

- Digital Sensor

→ Responses in binary nature.
→ Designs to overcome the disadvantages of analog sensors.
→ Along with the analog sensor it also comprises of extra electronics for bit conversion.

Example: Passive infrared (PIR) sensor and digital temperature sensor (DS1620)

- Scalar Sensor

→ Detects the input parameter only based on its magnitude.

→ The response of the sensor is a function of magnitude of the input parameter.

→ Not affected by the direction of the input parameter.

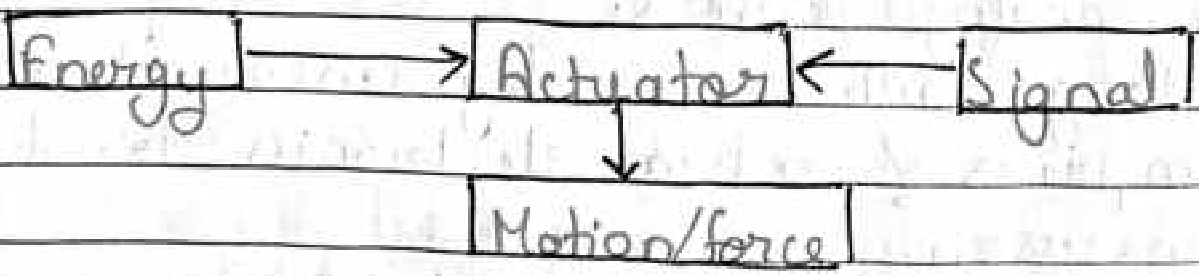
Example - Temperature, gas, strain, color, and smoke sensors.

- Vector Sensor

The response of the sensor depends on the magnitude of the direction and orientation of input parameter.

Example- Accelerometer, gyroscope, magnetic field, and motion detector sensors.

Actuator



- An actuator is part of the system that deals with the control action required

(mechanical action)

- Mechanical or electro-mechanical devices.

- A control signal is input to an actuator and an energy source is necessary for its operation.
- Available in both micro and macro scales (sizes)

Example - Electric motor, solenoid, hard dri
comb drive, stepper motor.

Classification of Actuators

→ Electric Linear
→ Electric Rotary
→ Fluid Power Linear
→ Fluid Power Rotary
→ Linear Chain Actuators
→ Manual Linear
→ Manual Rotary

- Electric Linear

→ Powered by electric signal
→ Mechanical device containing linear gu
motors, and device mechanisms.
→ Converts electrical energy into Linear

displacement.

→ Used in automation applications including electric bell, opening and closing dampers, locking doors,

- Electric Rotary Actuator

→ Powered by electrical signal

→ Converts electrical energy into rotational motion.

→ Applications including ~~gro~~ quarter turn ~~valves~~ values. windows and robotic

- Fluid Power linear Actuator

→ Powered by hydraulic fluid, gas or differential air pressure.

→ Mechanical devices have cylinder and piston mechanisms.

→ Produces linear ~~d~~ displacement.

→ Primarily used in automation application including clamping and welding.

- Fluid Power Rotary Actuator

→ Powered by fluid, gas, or.

→ Consisting of gearing and cylinder an piston mechanisms.

→ Produces rotational motion.

→ Primarily applications of this actuator are opening and closing dampers doors and clamping

- Linear Chain Actuator.

→ Mechanical devices containing sprockets and sections of chain.

→ Provides linear motion by the free ends of the specially designed chains

→ Primarily used in motion control applications.

- Manual Linear Actuator

→ Provides linear displacement through the translation of manually rotated screws and gears.

→ Consists of gearboxes, and hand operated Knobs or wheels.

→ Primarily used for manipulating tools and work pieces.

- Manual Rotary Actuator

→ Provides rotary output through the translation of manually rotated screws, levels or gears.

→ Consists of hand operated Knobs, levers, hand wheels and gear boxes.

→ Primarily used for the operation of valves

Humidity Sensors (hygrometer).

It senses, measures and reports both moisture and air temperature. The ratio of moisture in the air to the highest amount of moisture at a particular air temperature is called relative humidity.

It work by detecting changes that alter electrical currents or temperature in the air.

Types → Capacitive
→ Resistive
→ Thermal

• Capacitive

→ It measures relative humidity by placing a thin strip of metal oxide between two electrodes.

→ The metal oxide's electrical capacity changes with the atmosphere's relative humidity.

→ Weather, commercial and industries are the major application areas.

• Resistive

→ It utilize ions in salts to measure the electrical impedance of atoms. As humidity changes, so do the resistance of the electrodes on either side of the salt medium

• Thermal

→ Two thermal sensors conduct electricity based upon the humidity of the surrounding air.

→ One sensor is encased in dry nitrogen while the other measures ambient air.

→ The difference between the two measures the humidity.

Working.

It usually contain a humidity sensing element along with the thermister to measure temp. (Types).

Applications.

→ It is used for various applications for measuring humidity in HA HVAC systems, Printers, Fax machines, Weather stations, automobiles, food processing, refrigerators etc.

→ Due to there low cost and small size, resistive sensors are used in residential, industrial and domestic applications.

→ Thermal conductors are commonly used in pharmaceutical plants, food dehydration, drying machines etc.

Temperature Sensor

It is a device, used to measure the temperature through an electrical signal. it requires a thermocouple or RTD(Resistance Temperature Detectors).

Working

The measurement of the temperature sensor is about the hotness or coolness of an object. The working base of the sensors is the voltage that read across the diode. If the voltage increases, then the temperature rises and there is a voltage drop between the transistor terminals of base & emitter, they are recorded by the sensors.

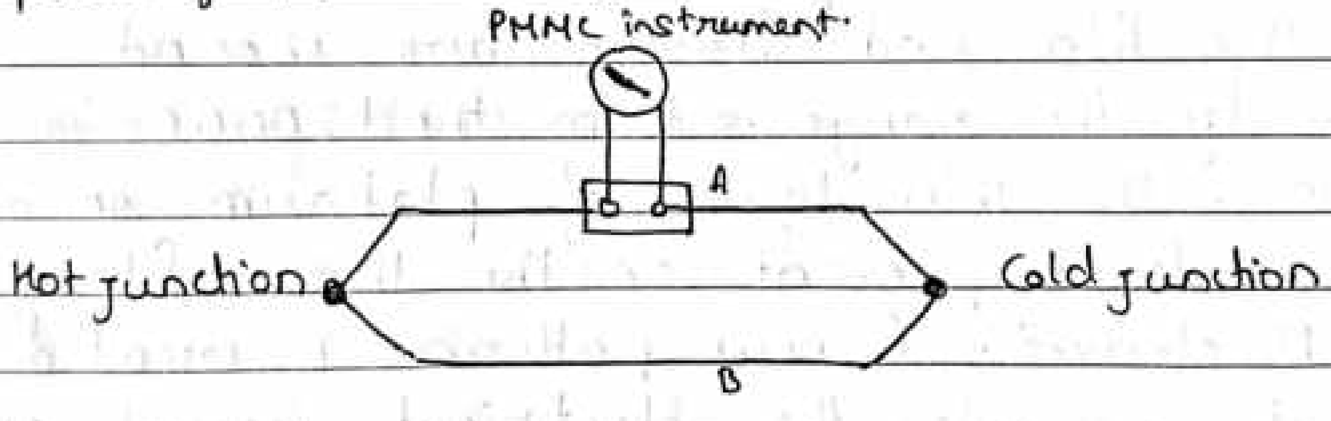
If the difference in voltage is amplified, the analogue signal is generated by the device and it is directly proportional to the temperature.

Types of T.S

There are many different types of T.S

- Thermocouple Sensor

~~A temperature Sensor is the instrumentation~~
A thermocouple is a temperature-measuring device consisting of two dissimilar conductors that contact each other at one or more points. It produces a voltage when the temperature of one of the points differs from the reference temperature at other parts of the circuits.



- Thermistor Sensor.

This type of sensors is used mostly in the human thermometers. If there is a change in the temperature, then the electrical current or resistance also changes. The thermistor is prepared by using the semiconductor materials with a resistivity which is especially sensitive to temperature. The resistance of a thermistor decreases with increasing temperature so that when the temperature changes, the resistance change is predictable.

- Resistance Temperature Detector

These are the temperature sensors with a ~~sust~~ resistor that changes the resistive value simultaneously with temperature changes. The RTDs are used in a wide temperature range from -500C to 5000C for thin film and for the wire wound variety the range is from the +2000C to 8500C. The thin layer of platinum on a substrate is present on the thin film RTD element. A new pattern is created which provide the electrical circuit and it is trimmed to give a specific resistance.

- Thermometer

It is a device which is used to measure the temperature of any glass solids, or ligwids. In this type or alcohol is used in a tube whose volume is changed by changing the temperature. Its volume is directly proportional to temperature.

- IR Temperature Sensor

These are an electronic and non-contacting sensor which have a certain characteristics such that it can de emits the IR radiations. Two types of IR T.S. used in market are IR S and Quantum IR.S. It detects the surface temperature by emitting radiations. Thus its cost depends on its working capabilities ~~meas~~ means its accuracy level depends upon its cost in other words low cost - low accuracy level and high cost - high accuracy level.

- Semiconductor based Sensor / ICs T.S.

It operate with reverse bias, have a small capacitance and a low leakage current. They are formed on thin wafers of silicon. They are compact, produce linear outputs, and have a small range of temperature. They also have low cost and are accurate following calibration.

Types:- 1) Voltage output ic
2) Current Output

3) Digital output
4) Resistance output
5) Simple diodes.

Applications of T.S

1) These are used in electric motors for measuring the motor winding temp., bearing temp., brushes temperature.

2) These are used in electric cables for measuring the cable internal temperature.

3) In mechanical engines for measuring engine oil temp & engine bearing temp.

4) In rubber, plastic, biomedical industries.

Ult.

Ultrasonic Sensor:

An ultrasonic sensor is measures the distance of respective object by sending the wave of specific frequency. This sound wave is reflected after the collision with respective object and this wave is received by the ultrasonic receiver. Distance is measured by calculating sending and receiving time of this sound wave.

Distance = Sound speed x time taken /2.

Working

It consists of set of ultrasonic transmitte and receiver which are operated at same frequency. When anything or object comes into the area of covered circuit then its frequency sound reflected to receiver and alarm is triggered. This sensor circuit is very sensitive and it could be reset automatically or still in triggered until it is reset manually.

Types

- Ultrasonic Proximity Sensors.

A special type of sonic transducer is used in this sensor for alternate transmission and reception of sound wave. This sonic transducer emits the sonic waves which are reflected by an object and after this emission, this sensor switched into receive mode.

- Ultrasonic 2 Point Proximity Sensors Switches

It consists of 2 points for switching, therefore it is called 2-point proximity switches. It is almost similar with standard sensor only differ the 2-touch set up Key and this function is called Tech-in function. Its switches Sdal & Sda2 could be easily programmed within the sensing range with the help of built in Tech-in button.

- Ultrasonic Retro reflective Sensors:

The operation of ultrasonic retro reflective sensor is similar with ultrasonic proximity sensor.

Only difference; in this sensor the distance between sensor to reflector is measured by measuring the propagation time. In this sensor, the stationary object could be used as a reflector and sensing distance (SD) could be adjust by adjusting the potentiometer resistance with in ultrasonic sensor.

- Ultrasonic Through beam Sensors

Unlike proximity and retro-reflective sensors these sensors separate the emitter and the receiver into separate housings. The emitter sends a continous signal, which is then picked up by the receiver. When an object disrupts the sonic beam, the receiver reacts and triggers an output.

Arduino

→ Arduino is an open-source prototyping platform used for building electronics project

→ It consists of ~~a~~ both a physical programmable circuit board and a software, or IDE that runs ~~on~~ on your computer, where you can write and upload the ~~com~~ code to the physical board.

→ It The Arduino board adapting to the new needs and challenges, differentiating it from simple 8 bit boards to products for IoT applications, 3D printing

→ It can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet and even your smartphone or your TV.

Features

1) IDE runs on every ~~platform~~ operating system (Mac, Linux and Windows). B.

2) It's based on a strong and well supported backened, the open source gcc toolchain and wrapped in Java, so bugs can be found and fixed.

3) There is a big community of smart people using and working on the IDE to keep it going strong.

4) There are numerous object wrapped libraries to do complex things.

5) The code runs directly on bare metal, with a well tested and up understood compiler.

6) It became a huge hit. because of its analog to digital input.

7) It is easily affordable and there is no comprise with low quality board.

Arduino Variants

- Arduino Uno
- " NaNo
- " Lilypad
- " Mega 2560
- RedBoard.

Raspberry PI:

→ Raspberry Pi is a low cost, credit card ste sized computer that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python

→ It is a project initiated by Eben Upton and developed in UK by Raspberry Pi foundation in 2009.

→ It supports several programming languages like scratch, Python, Node.JS, C, C++, Java, Perl, HTML5, Javascript, JQuery etc.

2 It is capable of doing everything that a desktop computer can do.

Features & Benefits

1) It's simple, open and easy to maintain and energy efficient.

2) Small in size and at the same time has all the functions of a laptop and a desktop.

3) It consumes very less power, only about five to seven watts of electricity.

4) Systems are noise free and is a perfect adaptive technology where it is able to display images or play videos at 1080p HD resolution

5) It is very ~~com~~ affordable compared to branded computers that are commercially available

6) It is armed with built in HDMI capable graphics.

7) It can be overclocked if there are performance problems with the application used.

8) The ability to store an SD card makes it easy to swap with other SD cards.

- Lite Os

- It is a lightweight, open source IoT device and smartphone OS from the Chinese smartphone manufacturer Huawei.
- It is designed to have a low footprint, which saves space and reduces the load of the OS on the device.
- It supports smartphones, wearables, intelligent manufacturing applications, smart homes and Internet of Vehicles (IoV).
- It simplifies IoT device development and connectivity while focusing on enhancing user experience.
- The smallest Kernel (6 KB) on the market offers fast-start and low power consumption features.

RIoT Os

- Open source Embedded O.S.
- It is designed for networked and memory constrained systems.
- Targeting on low power and IoT devices.
- Lightweight, limited processing time, small main\memory

- First developed by FU Berlin, INRIA and the HAW Hamburg in 1999.
- Written in ASC ANSI C.
- Based on a ~~microcontroller~~ microKernel architecture.

<u>Features</u> :-

- Modularity
  → Customization of the system's configuration.
  → Minimized Kernel's size.
  → Effects of bugs is limited to the module itself.
- Tickless Scheduler
  → It does not have a timer that fires periodically in order to emulate concurrent execution by switching threads continuously.
- Straight forward interrupt handler.
- Support various hardware vendors.
- Reliability and real time features.
  → zero latency interrupt handlers.
  → minimum context switching times with thread priorities.
- Support for full multithreading and C++.
- Full support for internet protocols on resource constrained system.

Contiki Os

→ It is an open source O.S for the IoT. I

→ It connects tiny low-cost, low power microcontrollers to the Internet and provides powerful low power internet communication.

→ It supports full standard IPV6 and IPV4 along with the recent low + power wireless standards: 6lowpan, RPL, CoAP.

→ It uses a minimalist design while still packing the common tools of modern Os:

→

Features

1) It comes with a rich set of features that are doc programmer friendly.

2) It can fit into 10 KB of RAM and 100 KB of ROM.

3) It can run on devices such as 8051 SoC to ARM powered devices.

4) Ports are available on other platforms such as Arduino and Atmel.

5) It comes with much documentation apart from well documented code.

OS. Functions include:

1. Process management
2. Memory management
3. Communication management
4. File management.

Applications:

1) There are several app. that come packaged as part of Contiki like small web browser, Web server, calc., shell, email client, ftp etc.
2) Developers can find tools like Cooja simulator for app. development.
3) Power sensitive applications.

Tiny Os

- It is a free open source operating system.
- Designed for wireless sensor networks.
- Tiny OS began as a collaboration between Univery of California, Berkely and Intel Research.
- An embedded operating system written in nes C language.
- It features a component based architecture.

Features

- Completely non blocking
- Programs are built out of software components.
- Tasks are non preemptive and run in FIFO order.
- Tiny OS code is statically linked.
- Power efficient as it makes the ~~a~~ sensor sleep as soon as possible.
- Component based architecture allows frequent changes while still keeping the size of code minimum.
- Event based execution model means no user / Kernel boundary and hence supports high concurrency.

Models

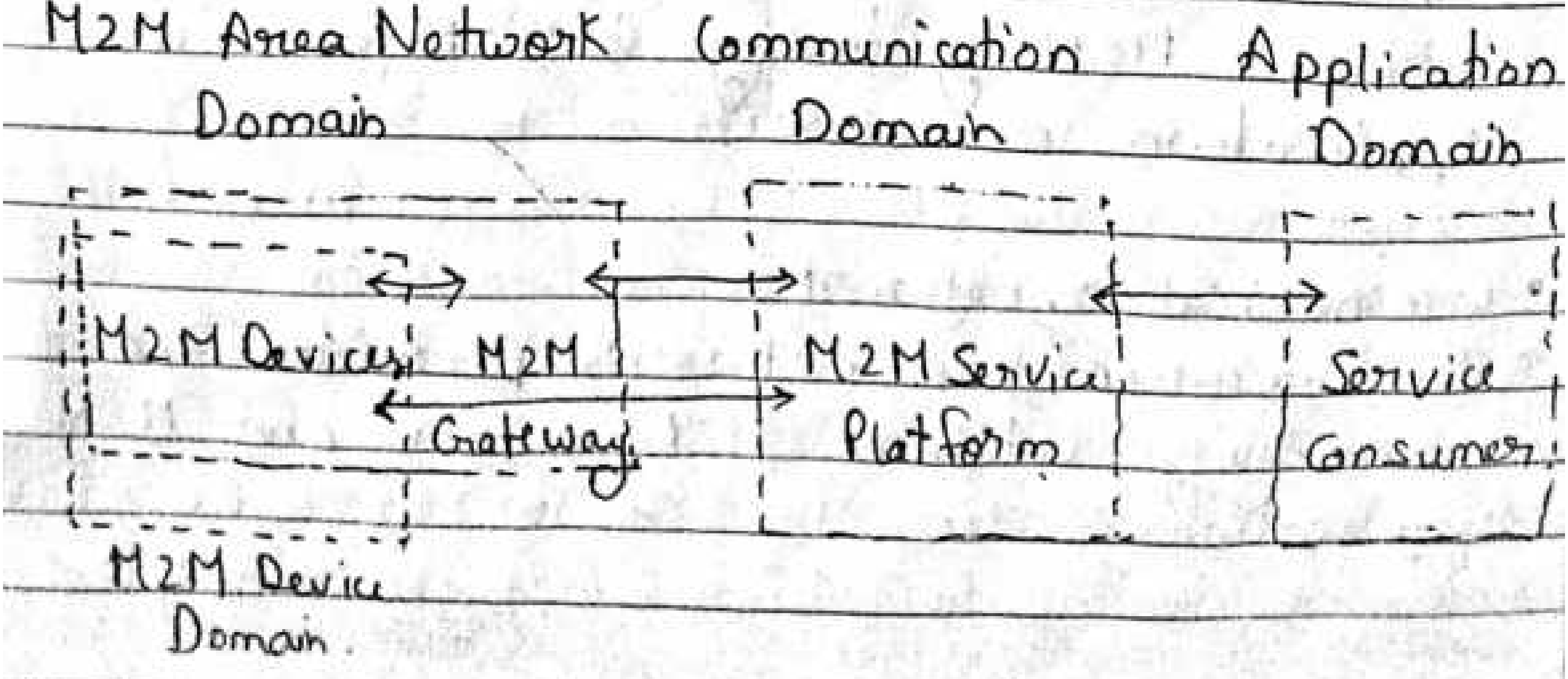
1) Data
2) Thread
3) Programming
4) Component
5) Network. ⟹

M2M

- Machine to Machine (M2M) refers to the communication or exchange of data betwee to two or more machines without human interfacing or interaction.
- Communication in M2M may be wired or wireless systems.
- Hu M2M uses a device such as sensor, RFID, meter, etc. to capture an 'events' like temp. inventory level, etc. that translates the captu event into meaningful information.
-

M2M System Architecture

1. M2M area networks
2. Communication networks
3. Application domains.
4. M2M gateways.



- M2M area networks

→ M2M network area consists of machines or M2M nodes which communicate with each other. The M2M nodes embedded with hardware modules such as sensors actuators and communication devices.

→ M2M uses communication protocol such as Zigbee, Bluetooth, Power line communication (PLC) etc.

→ M2M nodes communicate with in one network it can't communicate with external network node.

- M2M Gateways.

→ The Gateway module provides control and localization services for data collection

→ M2M communication network serves as infrastructure for realizing communication between M2M gatway & M2M end user application or server.

- Communication networks.

→ The communication network provides the connectivity between M2M nodes and M2M applications.

→ It uses wired or wireless network such as LAN, LTE, WiMAX, satellite communication etc.

• Application domains.

→ It contains the middleware layer where data goes through various app. services and is used by the specific business processing engines.

→ Applications may either target at end users, such as user of a specific M2M solution, or at other application providers to offer more refined building blocks by which they can build more sophisticated M2M solutions & services.

Difference between IoT and M2M

M2M	IoT
• Machine to machine communication and completly base hardware based	• Machine to machine, M to sensors, or humans to machines and software based.
• It is a point to point communication and uses non IP protocols	• It uses IP networks & protocols as the ~~communicat~~ communication is multipoint.

• These devices don't rely on Internet.	• Devices required internet connections.
• Data can be stored locally.	• Data can be stored locally and also in cloud.
• Limited integration option devices must have corresponding communication standards	• Unlimited integration option, but requires a solutions that can manage all the communication.
• Unidirectional comm.	• Bidirectional comm.

Similarities between IoT & M2M

Both provide remote ~~acess~~ access to machine data and both exchange info among machines without human intervention

Software Defined Networking (SDN)

- SDN is defined as the physical separation of the networking architecture of control plane from the data plane, and centralizes the network controller.

Basic Concepts of SDN

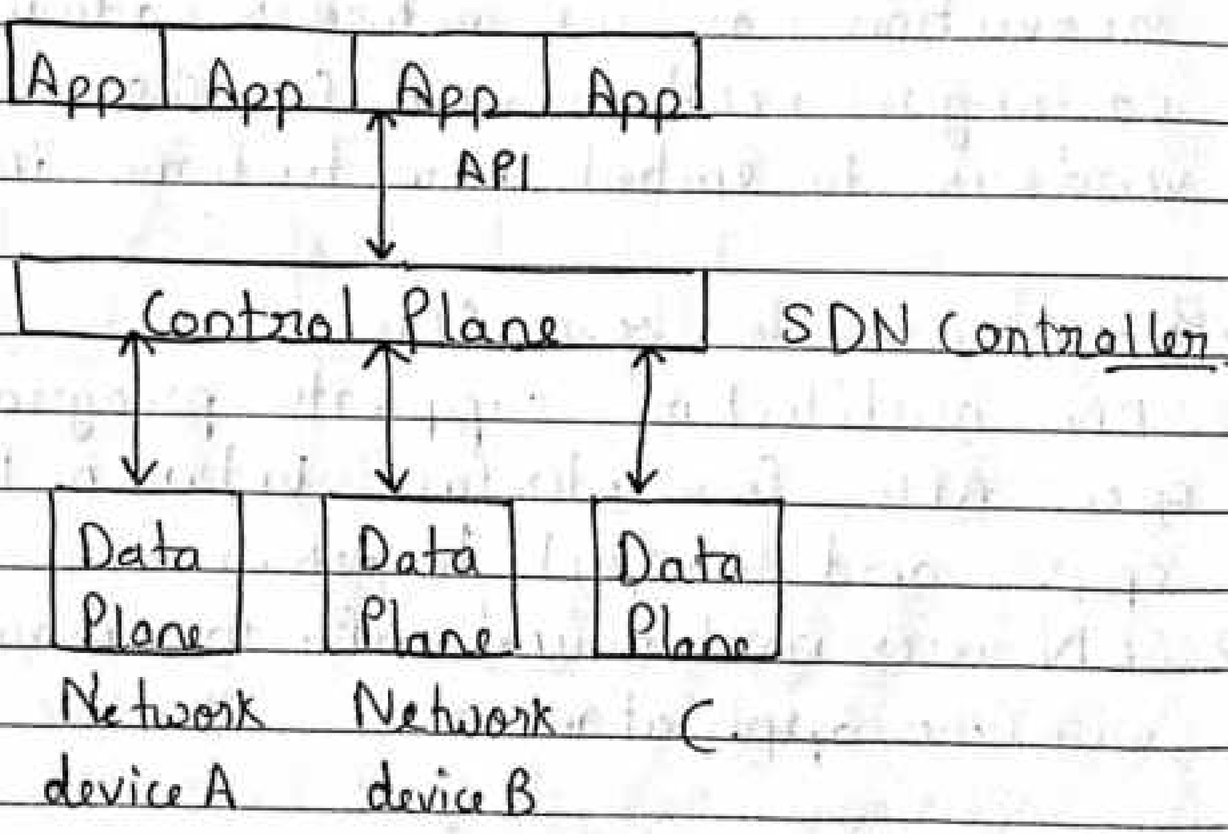
→ separate control logic from hardware switches

→ Define the control logic in a centralized manner.

→ Control the entire network including individual switches.

→ Communication between the app., control, and data planes are done through APIs.

SDN Architecture



Key Components of SDN.

- Centralized Network Controller.
- Programable open APIs.
- Standard communication interface (Open Flow).

1) Centralized Network Controller.

→ With separated control plains data, plane and centralized network controller, the network administrator can rapidly configure the network.

→ SDN application can be deployed through programmable APIs. which speeds up innovation as the network administrator no longer need to wait for the device vendors to embed new features in hardware

2) Programmable open APIs

→ SDN architecture supports programmable open APIs for interface between the SDN app. and control layers.

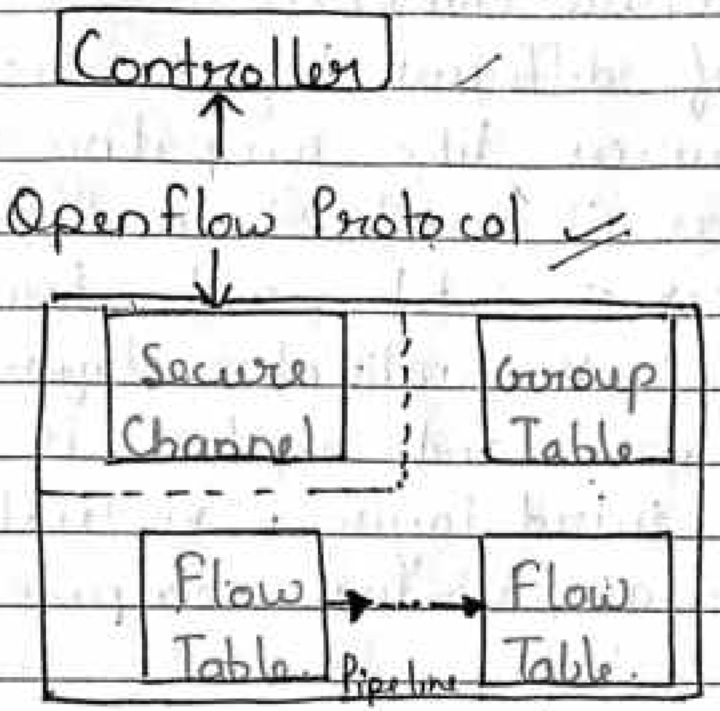
→ SDN uses northbound APIs to communicate with the applications.

3) Open Flow

→ Standard communication interface between control layer and infrastructure layer.

→ It uses Southbound APIs to relay information to the switches and routers below.

→ The controller manages the switch via open flow switch protocol where controller can add update and delete flow entries in ~~flowte~~ flow table.



OpenFlow Switch.

Network function Virtualization (NFV)

- Network functions virtualization (NFV) is the concept of replacing dedicated network appliances such as routers and firewalls with software running on general purpose CPUs or virtual machines, operating on standard Servers.
- NFV providers the ~~infras~~ infrastructure on which SDN can run. NFV and SDN are mutually beneficial to each other but not dependent.

Key elements of NFV

1. NFV infrastructure (NFVI):-

- NFVI consists of different layers such as hardware resources like computing resources, storage resources (hard-disc), and network resources (routers, switch, and firewalls)
- Second layer is Virtualization layer which separates hardware and replaces it with software and third layer is virtualized resources such as virtual compute, network and storage.

2. Virtualized network function (VNF)

VNF is a software implementation is a network function which is capable of running over the NFV infrastruce (NFVI).

Ex- vfirewall, vRouters.

3. NFV management and orchestration :-

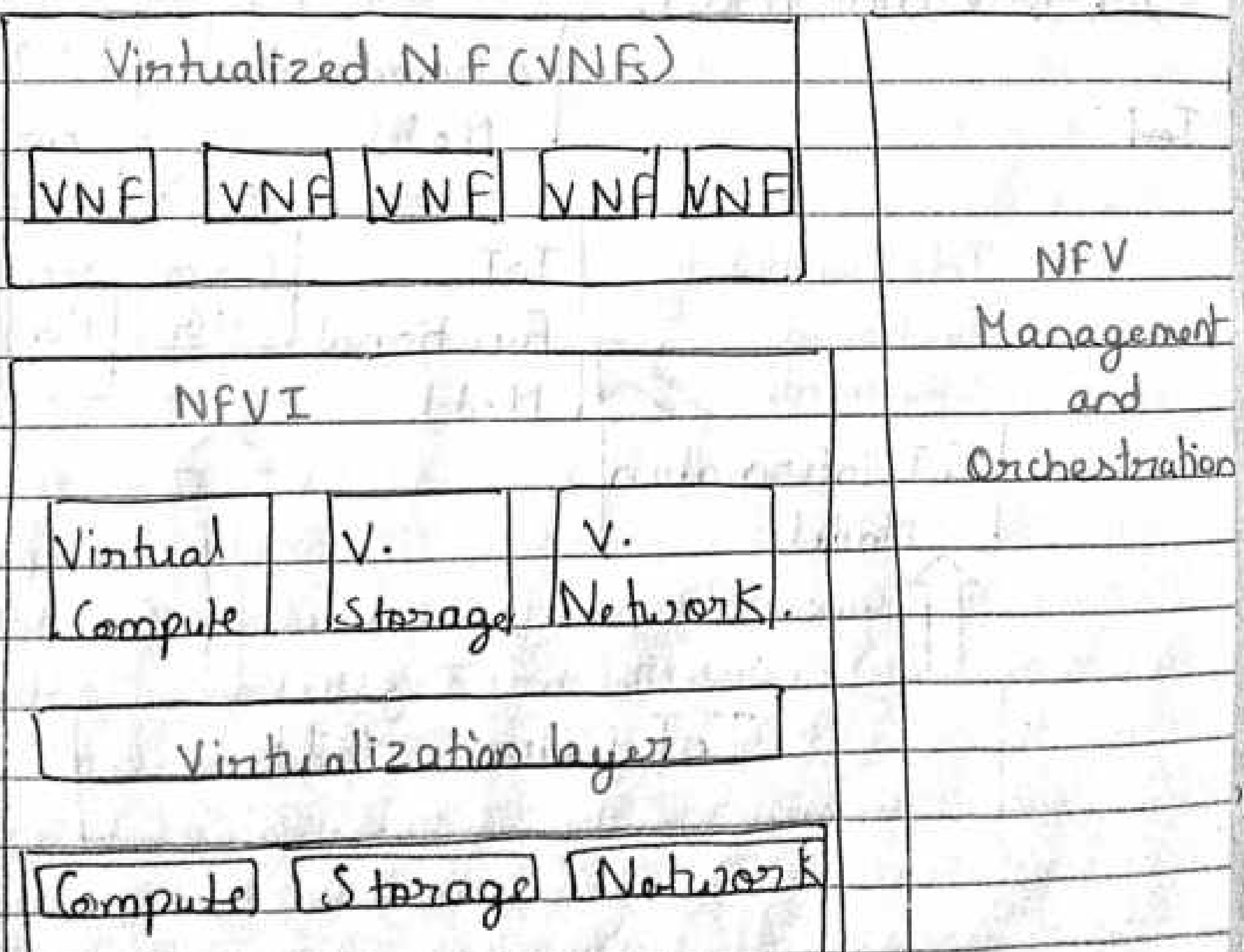
It has three parts.

→ Virtualized infrastructure manager:-

It controls and manages network functions with NFVI resources and monitors virtualization layer.

→ VNF manager:- It manages the life cycle of VNF such as initialize, update, query, scale, terminate. etc.

→ Orchestrator:-
It manages the life cycle of network services which includes policy management performance measurement and monitoring.



IoT Reference model and Architecture.

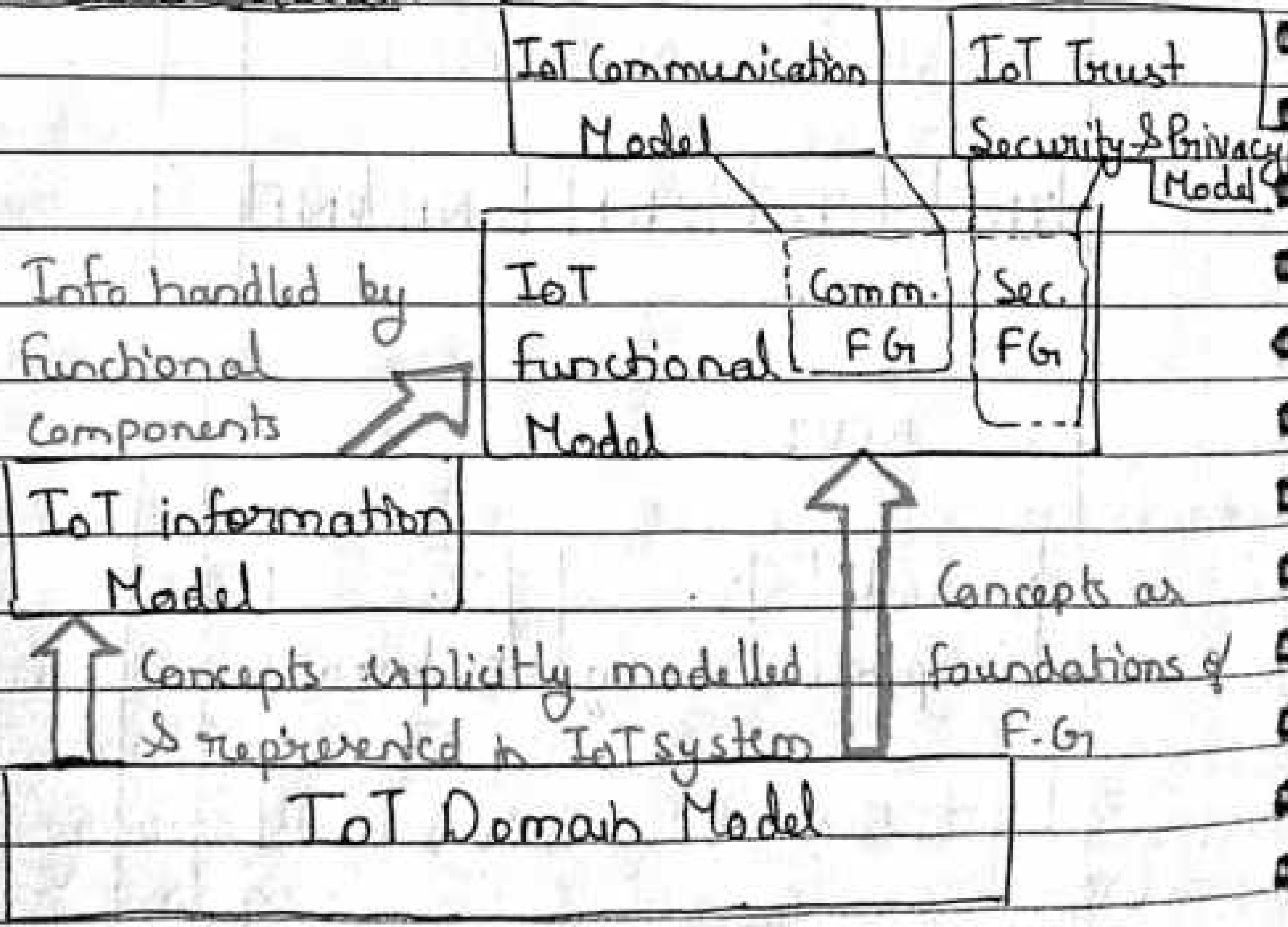
- An ARM consists of two main parts:

1/ a Reference model
2/ a Reference Architecture.

- A reference model describes the domain using a number of sub models.

1/ IoT Reference Model.

IoT


IoT Communication Model
IoT Trust Security & Privacy Model
Info handled by Functional Components
IoT Functional Model
Comm. FG
Sec. FG
IoT information Model
Concepts explicitly modelled & represented in IoT system
Concepts as foundations of F.G
IoT Domain Model

IoT domain model

→ It captures the basic attributes of the main concepts and the relationship between these concepts.

→ Abstraction level of the IoT Domain model has been chosen in such a way that its concepts are independent of specific technologies and use cases.

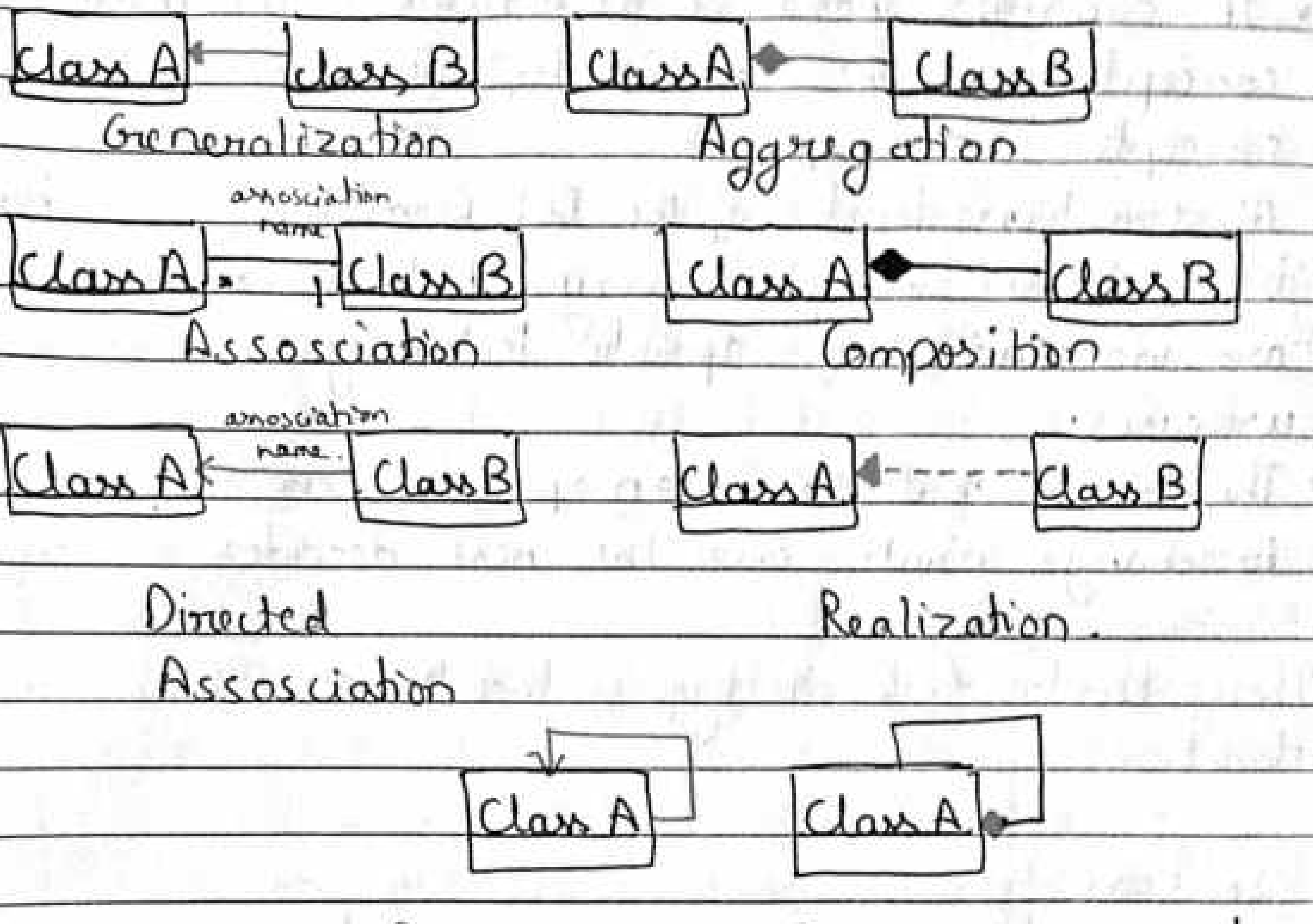
→ The idea is that these concepts are not expected to change much over the next decades or longer

Three Kinds of Device types for the IoT Domain Model.

1. Sensors
2. Actuators.
3. Tags. - In general identify the Physical Entity that they are attached to. It can be both devices or physical entities but not both, as the domain model shows.

   Example: Tag as a device - Radio Frequency ID.
   Tag as a P.E - Paper printed immutable barcode or Quick Response (QR) code.

Model notation and Semantics



Generalization

Aggregation

Association

Composition

Directed Association

Realization.

Reflexive D.A

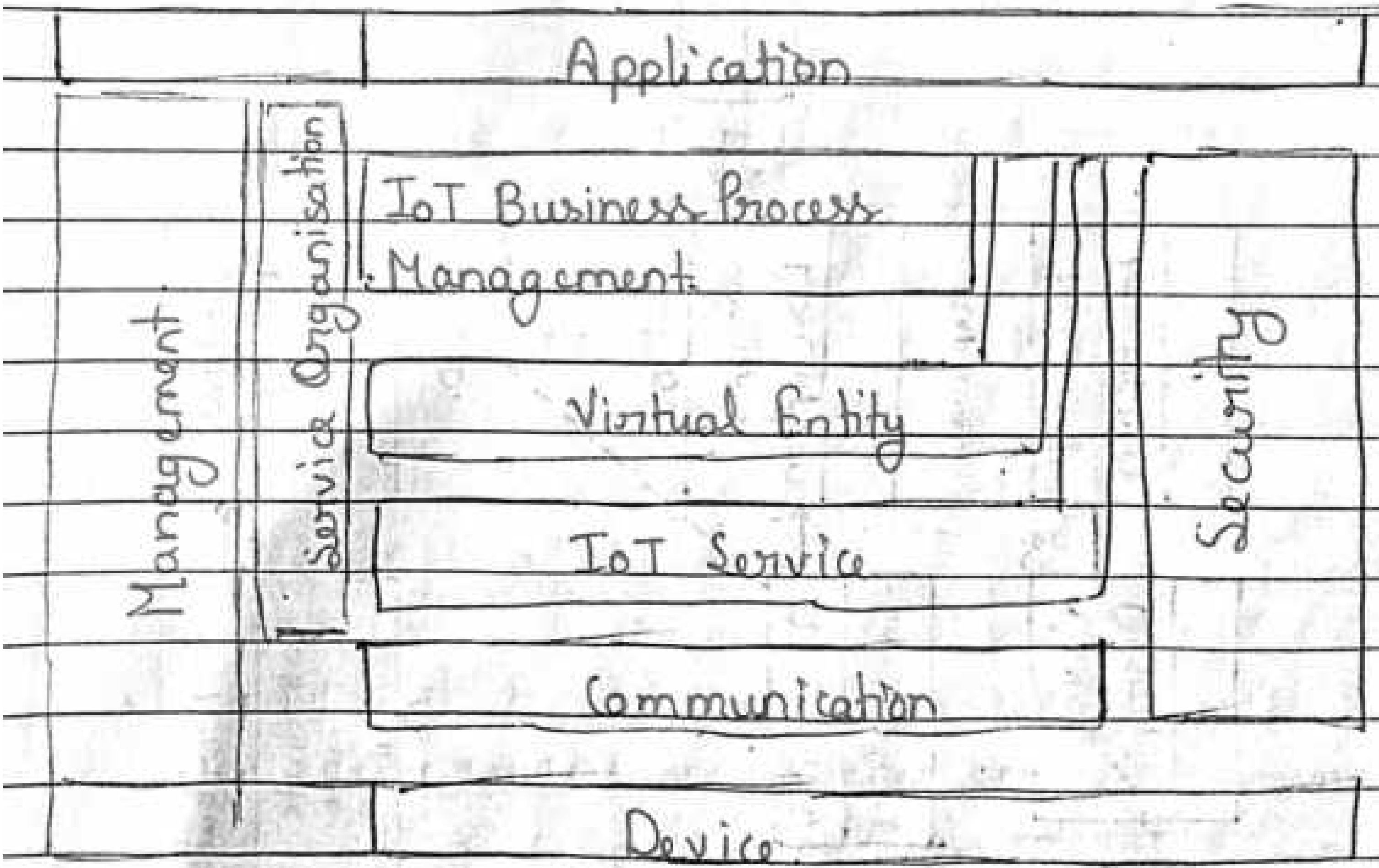
Reflexive Aggregation

IoT Information Model

Virtual entity in the IoT Domain Model is the 'thing' in the IoT, the IoT information model captures the details of a Virtual entity centric model. Similar to the IoT domain model, the IoT information Model is presented using Unified Modelling Language (UML) diagrams.

Functional model

>It aims at describing mainly the F.Gs and their interaction with the ARM, while the Functional View of a Reference Architecture describes the functional components of a~~n~~ FGs, interfaces, and interactions between the components. The functional View is typically derived from the functional Model is in conjunction with high-level requirements.



- Device functional Group.

-> The Device FGs contains all the possible functionalit hosted by the physical Devices that are used.

for increment the Physical Entities.
→ The Device functionality includes sensing actuation, processing, storage, and identification components, the sophistication of which depends on the Device capabilities.

- Communication functional group.

→ Comm. F.G. consists abstracts all the possible communication mechanisms used by the relevant Devices in an actual system in order to transfer information to the digital world components or other Devices.

- IoT Service F.G.

It corresponds mainly to the Service class from the IoT Domain model, and contains Single IoT services exposed by Resources hosted on Devices or in the network.

- Virtual Entity F.G.

→ It corresponds to the virtual entity class in the IoT Domain model
→ It contains the necessary functionality to manage associations between virtual Entities with themselves as well as as between V.E and related IoT Services.

IoT Service Organization functional group.

→ Its purpose is to host all functional components that support the composition ~~and of~~ of IoT and Virtual Entity services.

IoT Process Management F.G

⇒ It is a collection of functionalities that allows smooth integration of IoT related services with the business process.

Management F.G.

It includes the necessary functions for enabling fault and performance monitoring of the system, configuration for enabling the system to be flexible to changing user demands;

Security F.G.

It contains the functions that ensure the secure operation of the system as well as the managent of privacy. It components contains components for Authentication of Users, Authorisation of access to services by Users, secure communication between entities of the system such as Devices, Services, App..

Application F.G. - ..

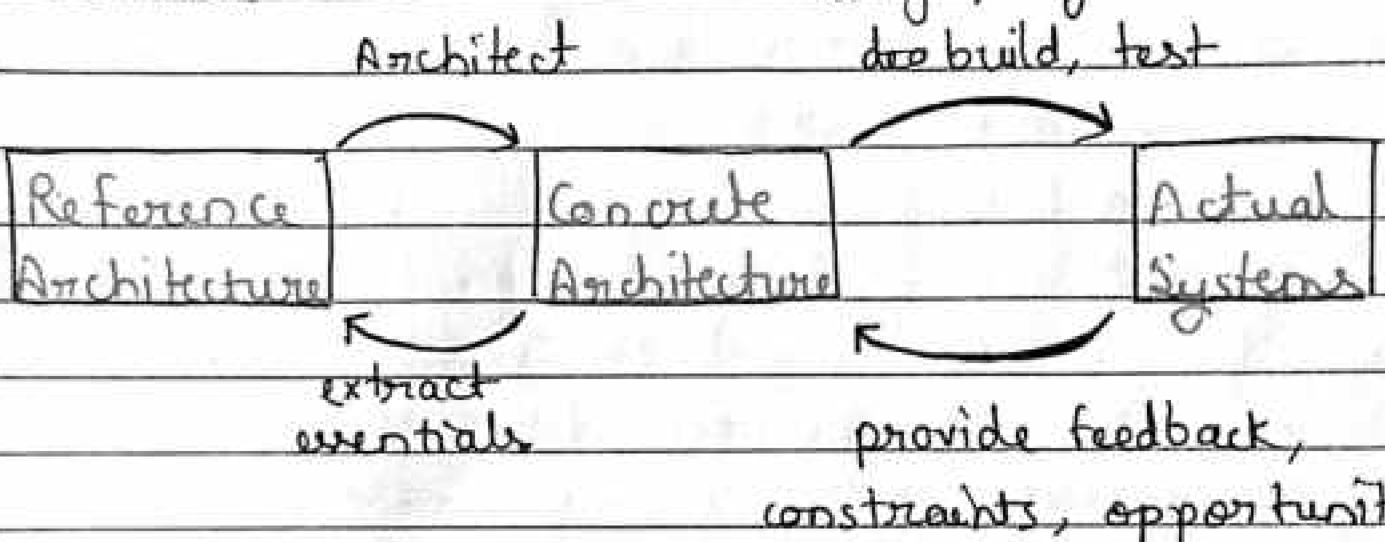
Communication Model.

It aims at defining the main communication paradigms for connecting elements, as defined in the IoT Doman Model.

IoT Reference Architecture

→ It is a starting point for generating concrete architectures and actual systems.

→ A reference architecture, serves as a guide for one or more ~~or~~ concrete system architets.



• Architectural Views

→ It is presented as sot of architechural views. Views are useful for reducing the complextity of the Reference.

→ Views are used during the design and implementation phase of a concrete system architecture.

→ A view is composed of viewpoints which is a collection of patterns, templates and ~~con~~ conventions for constructing one type of view.

- Functional view

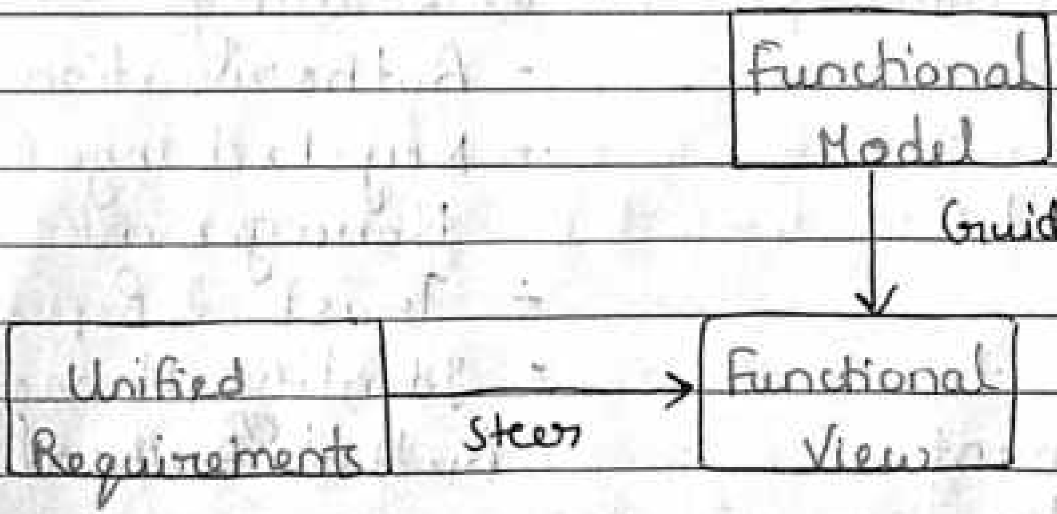
→ Describes ~~&~~ what the system does and its main functions.

→ The Unified Requirements are mapped to the diff. Functionality Groups of the IoT Functional Model.

→ Next, clusters of requirements of similar functionality are formed and a Functional Component for these requirements defined.

→ Thus the view points used for constructing IoT functional View are :-

1) Unified Requirements
2) IoT Functional Model



Functional view Process diag.

→ Once all functional components are defined the default function set, system use cases, sequence charts and interface definitions are made.

→ Following are the functional components for each of the functionality groups.

1) IoT Process Management
- Process Modelling
- Process Execution

2) Service Organisation
- Service Composition
- Service Orchestration
- Service Choreography

3) Virtual Entity
- VE Resolution
- VE & IoT Service Monitoring
- VE Service

4) IoT Service
- IoT Service
- IoT Service Resolutio

5) Communication
- Hop to Hop "
- Network "
- End to End "

6) Security
- Authorization
- Key Exchange & Management
- Trust & Reputation
- Identity Management
- ~~Authod~~ Authentication

7) Management
- Configuration
- Fault
- Reporting
- Member
- State.

- Information View.

→ It describes the information that the system handles and the components that handle these information.

→ The pieces of information handled by an IoT system complying to an ARM such as the IOT A are the following:

- Virtual Entity context information i.e. attributes (simple or complex) as represented by parts of the IoT information model.

- IoT Service Output itself is another ~~impt~~ important part of information generated by an IoT system.

- Virtual Entity descriptions and its association with other Virtual entity.

- Resource descriptions - type of resources, identity, associated services and devices.

- Device descriptions like device capabilities

- Descriptions of Composed services like the model of how a complex service is composed of simpler services.

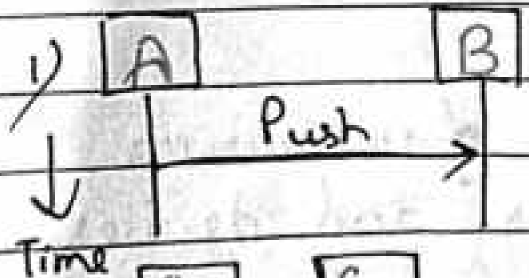
- IoT Business Process Model describes the steps of a business process utilizing other IoT related services.

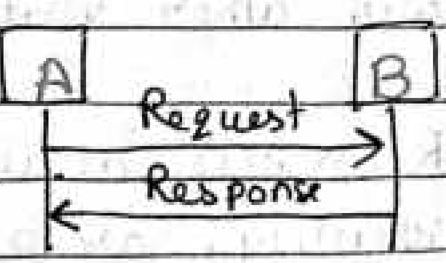
- Management information such as state information from operational FC

Information handling

-> The presentation of information handling in an IoT system assumes that FCs exchange and process information.

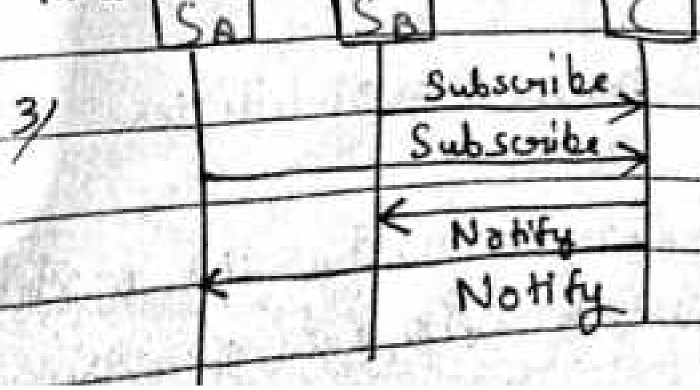
-> The exchange of information between FC follows the interaction patterns below



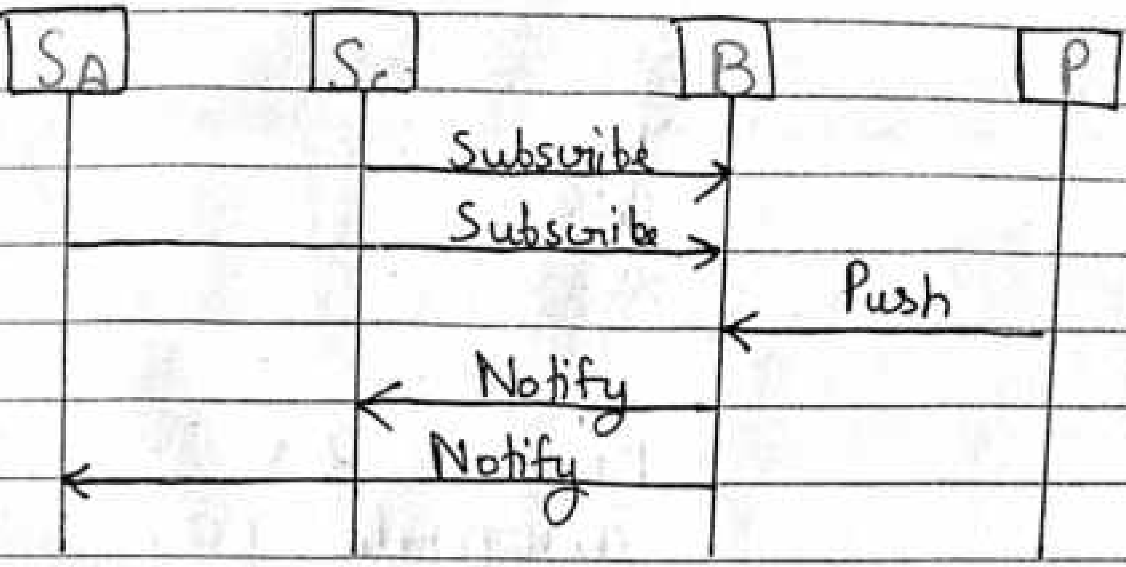


S_A — S_B

4) §



4)



- Deployment and Operational View.

→ Description of the main real world components of the system such as devices, network routers, servers etc.

→ It aims at providing users of the IoT Reference model with a set of guidelines to drive them through the different design choices that they have to face while designing the actual implementation of their services.

→ It will discuss how to move from the service & description and the identification of the different functional elements to the selection among the many available technologies in the IoT to build up the overall networking behaviour for the deployment.

Representational State Transfer (REST).

→ It is a type of software architecture that was designed to ensure interoperability between different internet computer systems. &

→ It works by putting in place very strict constraints for the development of web services.

→ Services that can request and edit text version of a web resource via a predefined set of operations that are uniform and stateless.

Architectural Constraints.

- Client - Server

→ Separation of concerns is the principle behind the Client-server constraints.

→ By separating the user interface concerns from the data storage concerns, we improve the portability of the user interface across multiple platforms and improve scalability by simplifying the server components.

- Stateless

→ This constraint states that the Server does not store any session data.

→ It means that all the information to understand a request is contained within the request.

→ Improves scalability.

→ Session state is therefore Kept entirely on the client.

- Cacheable

→ It requires that every response should include whether a response can be cacheable or not.

→ For subsequent requests, the Client can retrieve from its cache, to need to send request to the Server.

→ Reduces network latency, improves efficiency, scalability.

- Uniform interface

→ Uniform interface is the Key differentiator between REST & ~~Rf~~ Non REST APIs.

→ There are 4 elements of Uniform Interface constraint.

- Identification of Resources (typically by an URL)
- ~~Manage~~ Manipulation of Resources through representations
- Self-descriptive messages for each request.
- HATEOS (Hypermedia As the Engine of app. State)

→ Promotes generality as all components interact in the same way.

- Layered System

→ In or It allows an architecture to be composed of hierarchical layers.

→ Each layer doesn't Know anything beyond the immediate layer.

→ Disadvantage is latency.

- Code on Demand

→ It allows client functionality to be extended by downloading and executing code in the form of applets or scripts.

→ Allowing features to be downloaded after deployment improves system extensibility.

Architectural properties

- Scalability allowing the support of large numbers of components and interactions among components.
- Simplicity of a uniform interface.
- Modafibility of components to meet changing needs.

- Visibility of communication between components by service agents.
- Portability of components by moving program code with the data.
- Reliability in the resistance to failure at the system level in the presence of failures within components, connectors or data.

Uniform Resource Identifiers

- A URI is a sequence of characters that identifies a logical or physical resource. ~~URI are~~ ~~specified in the internet engineering~~
- It describes the mechanism used to access resources, the computers on which resources are housed and the names of the resources on each computer.

There are two types of URIs

- URL (Uniform Resource Locator)

→ It is the mechanism used by browsers to retrieve any published resource on the web.
→ It is the address of a given unique resource on the web.
→ It is handled by the webserver.

- Uniform Resource Name (URN)

→ URNs are globally unique persistent identifiers assigned within defined namespac so they will be available for a long period- of time, even after the resource which they identify ceases to exist or becomes unavailab

Challenges in IoT

- Security challenges in IoT

1. Lack of Encryption.

- Although encryption is a great way to prevent hackers from accessing data, it is also one of the leading IoT security challenges.
- These devices lack the storage and processing capabilities that would be found on a traditional computer.
- The result is an increase in attacks where hackers can easily manipulate the algorithms that were designed for protection.

2. ~~Out-dated legacy Security.~~

A

2. Insufficient testing and updating.

- With the increase in number of IoT devices, IoT manufacturers are more eager to produce and deliver their devices as fast as they can, without giving security too much of a thought.
- Most of these devices and IoT products don't get enough testing and updates and are prone to hackers and other security issues

3. Brute-forcing and the issue of default passwords

- Weak credentials and login details leave nearly all IoT devices vulnerable to password hacking and brute forcing.

Mirai Malware

- Any company that used factory default credentials on their devices is placing both their business and its assets and the customer and their valuable information at risk of being susceptible to a brute force attack.

4. IoT malware and ransomware.

- Increases with increase in devices.
- Ransomware uses encryption to effectively lock out users from serveral several devices and platforms. and steal user's valuable data info.
- For example - A hacker can hijack a computer camera and take pictures . .

By using malware access point, the hackers can demand ransom to unlock the device and return the data.

5. IoT botnets aiming at cryptocurrency.

- IoT botnets workers can manipulate data privacy, which could be a massive risk for an open crypto-market. The exact value and creation of cryptocurrencies could face danger from mal-intentioned hackers.
- ~~IoT~~ The blockchain companies are trying to boost security. Blockchain technology itself is not particularly ~~vr~~ vulnerable but the app development process is.

~~Des~~

- Design Challenges in IoT.

1. Battery life is a limitation.

Issues in packaging and integration of small size chip with low weight and lesser power consumption.

2. Increased cost and time to market

- Embedded systems are tightly constrained by cost.
- The need originates to derive better approaches when designing the IoT devices in order to handl the cost modelling or cost optimality with digital electronic components.
- Designers also need to solve the design time problem and bring embedded devices at the right time to the market.

3. Security of the system

- Systems have to be designed and implemented to be robust and reliable and have to be secure with cryptographic algorithms and security procedures.
- It involves different approaches to secure all the components of embedded systems from prototype to deployment.

Deveto

- Development challenges in IoT.

1. Connectivity

- It is the foremost concern while connecting devices applications and cloud platforms.

- Connected devices that provide useful front-end information is extremely valuable. But poor connectivity becomes a challenge where IoT sensors are required to monitor, process data and supply information.

2. Cross Platform Compatibility (~~Hardware &~~, Devices).

- IoT applications must be developed keeping in mind the technological changes of the future.
- It's development requires a balance of hardware and software functions.
- It is a challenge for IoT application developers to ensure that device and IoT platform delivers the best performance despite heavy OS, device updates and bug fixings.

3. Data Collection and Processing.

- In IoT development, data play an important role but what is more crucial here is the processing or usefulness of stored data.
- Along with security and privacy, development teams need to ensure that they plan well for the way data is collected, stored or

processed within an environment.

4. Lack of Skill set

- All of the development challenges above can only be handled if there is a proper skilled resource working on the IoT application developme
- A right talent will always get you past the major challenges and will be an important IoT application development asset.

Domain Specific IoTs

1. Home Automation

- Smart lighting

→ Smart lighting for homes helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or dimming the lights when needed.

→ Smart lighting solutions for more home achieve energy savings by sensing the human movements and their environments and controlling the lights accordingly.

- Smart Appliances.

→ Smart Appliances make the management easier and also provide status infor. to the users remotely.

→ Example 1) Smart washer/ dryer can be controlled remotely and notify when the washing / drying is complete.

2) smart Refrigerators can keep track of the items store and send updates to the users when an item is low on stock.

- Intrusion Detection.

→ Home intrusion detection systems uses security cameras and sensors (PIR) to detect intrusions and raise alerts.

→ Alerts can be in the form of an SMS or an email sent to the user.

→ Advanced systems can even send detailed alerts such as an image grab or short video clip.

→

- Smoke / Gas detectors

→ Smoke detectors are installed in home and buildings to detect smoke that is typically an early sign of fire.

→ It uses optical detection, ionization or air sampling techniques to detect smoke.

→ Gas detectors can detect the presence of harmful gases such as CO, LPG etc.

→ It can raise alerts in human voice describing where the problem is.

2. <u>cities</u>

- Smart Parking

→ It make the search for parking space easier and convenient for drivers.

→ These are powered by IoT systems that detect the no. of empty parking slots and send the information over the internet to smart parking application back-ends.

- Smart lighting

→ It allows lighting to be dynamically controlled remotely to configure lighting schedules and lighting intensity.

→ Custom lighting configurations can be set ~~of~~ for different situations such as a ~~fodd~~ foggy day, a festival etc.

→ Smart lights are equipped with sensors that can communicate with other lights and exchange information on the sensed ambient conditions to adapt the lighting.

- Smart Roads

→ Smart roads can provide info on driving conditions, travel time estimates and alerts in case of poor driving conditions, traffic congestions and accidents.

→ Such info can help in making the roads safer and help in reducing traffic jams.

- Structual Health Monitoring

→ This system uses a network of sensors to monitor the vibration levels in the structures such as bridges and buildings.

→ The data collected from these sensors is analyzed to ~~acces~~ assess the health of the structures {detecting cracks and mechanical breakdowns}, remaining life of the structure).

- Surveillance

→ Surveillance of infrastructure, public transport and events in cities is required to ensure safety and security.

→ City wide surveillance infrastructure comprising of large number of distributed and internet connected video surveillance cameras can be created.

- Emergency response.

→ IoT systems can be used for monitoring the critical infrastructure in cities such as buildings, gas and water pipelines, public transport and power stations.

→ Fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructure.

→ Such systems can reduce the latency of emergency services for vehicles such as ambulances and police cars while minimizing disruption of regular traffic.

3. Environment

• Weather Monitoring

→ W.M systems can collect data from a number of sensor attached (such as temp., humidity, pressure etc.) and send the data to cloud-based applications and storage back-ends.

→ The data collected in the cloud can then be analyzed and visualized by cloud based applications.

→ Weather alerts can be sent to the subscribed users from such applications.

• Air Pollution Monitoring

→ IoT based air pollution monitoring systems can monitor emission of harmful gases. by factories and automobiles using gaseous and meteorological sensors.

→ The collected data can be analyzed to make informed decisions on pollutions control approaches.

- Noise Pollution Monitoring

→ This systems uses a no. of noise monitoring stations that are deployed at different places in a city.

→ The data on noise levels from the stations is collected on servers or in the cloud.

→ The collected data is then aggregated to generate noise maps.

→ Noise maps can help the policy makers in Urban planning and making policies to control noise levels near ~~resen~~ residential areas, schools & park

- Forest fire detection.

→ Early detection of forest fires can help in minimizing the damage caused by forest fires.

→ IoT based forest fire detection systems use a no. of monitoring nodes deployed at different locations in a forest.

→ Each monitoring node that collects measurements on ambient conditions including temp., humidity, light levels etc.

- River floods Detection.

→ It can cause extensive damage to the natural and human resources and human life.

→ IoT based river flood monitoring system use a no. of sensor nodes that monitor the water level and flow rate.

→ Monitoring applications raise alerts when rapid increase in water level & flow rate is detected.

4. Energy

- Smart Grids.

→ Smart Grid is a data communications network integrated with the electrical grid that collects and analyzes data captured in near real time about power transmission, distribution and consumption.

→ By using IoT based sensing and measurement technologies the health of equipment and the integrity of the grid can be evaluated.

→ Smart meters can capture almost real time consumption, remotely control the consumption of electricity and remotely switch off supply when required.

- Renewable Energy Systems.

→ Due to the variability in the O/P from renewable energy sources integrating them into the grid can cause grid stability and reliability problems.

→ Variable output produces local voltage swings than can impact power quality.

→ When distributed renewable energy sources are integrated into the grid, they create power bi-directional power flows for which the grids were not originally designed.

→ IoT based systems at the point of interconnection measure the electrical variables and how much power is fed into grid.

- Prognostics

→ Energy systems have a large no. of critical components that must function correctly so that the system perform their operation correctly.

- Energy systems have thousands of sensors that gather real-time maintenance data continuously for condition monitoring and failure prediction purposes.

→ IoT based prognostic real-time health management systems can predict performance of machines or energy systems by analyzing the extent of deviation of a system from its normal operating profiles.

5. Retail

- Inventory Management.

→ Overstocking of products can result in additional storage expenses, understocking can lead to loss of revenue.

→ IoT systems using Radio frequency Identification (RFID) tags can help in inventory management and maintaining the right inventory levels.

- Smart Payments.

→ Smart payments solutions such as contact less payments powered by technologies such as Near field communication (NFC) and bluetooth.

→. Customers can store the credit card information in their NFC-enabled smart phones and make payments by bringing the smart phones near the point of sales terminals.

- Smart Vending Machines.

→ Smart vending machines connected to the Internet allow remote monitoring of inventory levels, elastic ~~pcia~~ pricing of products, promotions, and contact-less payments using NFC.

6) Logistics.

- Route Generation and scheduling

→ Route generation and scheduling systems can generate end to end routes using combination of route patterns and transportation modes and feasible schedules based on the availability of vehicles.

→ As the transportation network grows in size and complexity, the no. of possible route combinations increases exponentially.

→ IoT based systems backed by the cloud can provide fast response to the route generation queries and can be scaled up to serve a large transportation network.

- Fleet Tracking

→ Vehicle fleet tracking systems use GPS technology to track the locations of the vehicles in real-time.

→ Cloud based fleet tracking systems can be scaled up on demand to handle large no. of vehicles.

→ Alerts can be generated in case of deviations in planned routes.

- Shipment Monitoring

→ IoT based shipment monitoring systems use sensors such as temp., pressure, humidity, for instance to monitor the conditions inside the containers and send the data to the cloud, where it can be analyzed to detect food spoilage:

- Remote Vehicle Diagnostics.

→ This system can detect faults in the vehicles or warn of impending faults.

→ These diagnostic systems use on-board IoT devices for collecting data on vehicle operation and status of various vehicle sub-systems.

→ Such data can be captured by integrating on-board diagnostic systems with IoT devices using protocols such as CAN bus.

7) Agriculture.

- Smart Irrigation.

→ Smart irrigation systems use IoT devices with soil moisture sensors to determine the amount of moisture in the soil and release the flow of water through the irrigation pipes only when the moisture levels go below a predefined threshold.

- Green house control

→ The climatological conditions inside a green house can be monitored and controlled to provide the best conditions for growth of plants

→ The temperature, humidity, soil moisture, light and CO_2 levels are monitored using sensors and ~~the~~ are controlled automatically using actuation devices.

→ IoT systems play an important role in green house control and help in improving productivity

8) <u>Industry</u>

- Machine Diagnosis and Prognosis.

→ Machine Prognosis. - predicting the performance of a machine by analyzing the data on the current operating conditions.

→ Machine Diagnosis - determingging the cause of a machine fault.

→ Sensors in machines can monitor the operating conditions such as temp. and vibration levels.

→ ~~IoT~~

- Indoor Air Quality Monitoring

→ Monitoring indoor air quality in factories is important for health and safety of the workers.

→ IoT based gas monitoring systems can help in monitoring the indoor air quality using various gas sensors.

→ Wireless sensor networks based IoT devices can identify the hazardous zones, so that corrective measures can be taken to ensure proper ventilation.

v• Health and lifestyle.

- Health and fitness monitoring

→ twea Wearable IoT devices. that allow non-invasive and continuous monitoring of physiological parameters can help in continuous health and fitness monitoring.

→ These wearable devices may can be in various forms, Such as belts and wrist bands.